New Constructions of Statistical NIZKs: Dual-Mode DV-NIZKs and More

Benoît Libert, Alain Passelègue, Hoeteck Wee, David J Wu

New Constructions of Statistical NIZKs: Dual-Mode DV-NIZKs and More

Benoît Libert* Alain Passelègue† Hoeteck Wee‡ David J. Wu§

Abstract

Non-interactive zero-knowledge proofs (NIZKs) are important primitives in cryptography. A major challenge since the early works on NIZKs has been to construct NIZKs with a *statistical* zero-knowledge guarantee against unbounded verifiers. In the common reference string (CRS) model, such "statistical NIZK arguments" are currently known from k-Lin in a pairing-group and from LWE. In the (reusable) designated-verifier model (DV-NIZK), where a trusted setup algorithm generates a reusable verification key for checking proofs, we also have a construction from DCR. If we relax our requirements to *computational* zero-knowledge, we additionally have NIZKs from factoring and CDH in a pairing group in the CRS model, and from nearly *all* assumptions that imply public-key encryption (e.g., CDH, LPN, LWE) in the designated-verifier model. Thus, there still remains a gap in our understanding of statistical NIZKs in both the CRS and the designated-verifier models.

In this work, we develop new techniques for constructing statistical NIZK arguments. First, we construct statistical DV-NIZK arguments from the k-Lin assumption in *pairing-free* groups, the QR assumption, and the DCR assumption. These are the first constructions in pairing-free groups and from QR that satisfy statistical zero-knowledge. All of our constructions are secure even if the verification key is chosen maliciously (i.e., they are "malicious-designated-verifier" NIZKs), and moreover, they satisfy a "dual-mode" property where the CRS can be sampled from two computationally indistinguishable distributions: one distribution yields *statistical DV-NIZK arguments* while the other yields *computational DV-NIZK proofs*. We then show how to adapt our k-Lin construction in a pairing group to obtain new *publicly-verifiable* statistical NIZK arguments from pairings with a *qualitatively weaker* assumption than existing constructions of pairing-based statistical NIZKs.

Our constructions follow the classic paradigm of Feige, Lapidot, and Shamir (FLS). While the FLS framework has traditionally been used to construct computational (DV)-NIZK proofs, we newly show that the same framework can be leveraged to construct dual-mode (DV)-NIZKs.

1 Introduction

Non-interactive zero-knowledge (NIZK) proofs [BFM88, GMR89] allow a prover to send a single message to convince a verifier that a statement is true without revealing anything beyond this

*CNRS, Laboratoire LIP and ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, Inria, UCBL). Email: benoit.libert@ens-lyon.fr. Part of this research was supported by the French ANR ALAMBIC project (ANR-16-CE39-0006).

†Inria and ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, Inria, UCBL). Email: alain.passelegue@inria.fr.

‡CNRS, ENS, PSL. Email: wee@di.ens.fr. Supported in part by ERC Project aSCEND (H2020 639554).

§University of Virginia. Email: dwu4@virginia.edu. Part of this work was done while visiting ENS de Lyon. Supported by NSF CNS-1917414 and a University of Virginia SEAS Research Innovation Award.

fact. Although such NIZKs cannot exist in the plain model, they can be realized in the common reference string (CRS) model, where a trusted party generates and publishes a common reference string accessible to the prover and the verifier. Shortly after the introduction of NIZKs, numerous constructions have been developed in the CRS model from many classes of cryptographic assumptions such as factoring [BFM88, DMP87, FLS90, BY92, FLS99, DDO+01, Gro10, Gol11, GR13, CL18], pairing-based assumptions [CHK03, GOS06], and lattice-based assumptions [CCH+19, PS19]. We can also construct NIZKs in the random oracle model [FS86].

A major open problem since the early works on non-interactive zero-knowledge has been to construct NIZKs with a *statistical* zero-knowledge guarantee against *computationally-unbounded* verifiers (i.e., "statistical NIZK arguments"). Here, we only have constructions from the k-Lin family of assumptions over pairing groups [GOS06, GOS12] and LWE [PS19] (or circular-secure FHE [CCH+19]). If we relax the model and consider (reusable) designated-verifier NIZKs (DV-NIZKs), where the trusted party that generates the CRS also generates a *secret* verification key that is used to verify proofs, then the recent work of Chase et al. [CDI+19] provides an instantiation of a statistical DV-NIZK from the DCR assumption. In contrast, if we are satisfied with computational zero-knowledge, then we can additionally construct publicly-verifiable NIZKs in the CRS model from QR [BFM88], factoring [FLS99], and the CDH assumption over a pairing group [CHK03]. In the designated-verifier model, a recent line of works [QRW19, CH19, KNYY19a, KNYY19b, LQR+19] has provided constructions of computational DV-NIZKs from essentially all cryptographic assumptions known to imply public-key encryption. These include assumptions like CDH in a pairing-free group and LPN. Thus, there is still a gap in our understanding of statistical NIZKs in the CRS model, and especially in the designated-verifier model. In this work, we develop new techniques for constructing statistical NIZKs in both the standard CRS model as well as the (reusable) designated-verifier model, which we review below.

Reusable designated-verifier NIZKs. A key focus in this work is the designated-verifier model [PsV06, DFN06], where a trusted party generates the CRS together with a *secret* verification key that is used to verify proofs. In this work, we focus exclusively on *reusable* (i.e., multi-theorem) security where soundness holds even against a prover who has oracle access to the verification algorithm. We also consider the stronger *malicious-designated-verifier* model (MDV-NIZKs) introduced by Quach et al. [QRW19], where a trusted party only samples a common reference string,[1] and the verifier is allowed to choose its public and secret key-pair, which is used to generate and verify proofs, respectively. Here, we require that zero-knowledge should hold even if the verifier samples its public key maliciously. As discussed in [QRW19], MDV-NIZKs are equivalent to 2-round zero-knowledge protocols in the CRS model where the verifier's initial message is reusable. A recent line of works have shown how to construct (M)DV-NIZKs with *computational* zero-knowledge from nearly all assumptions known to imply public-key encryption (e.g., CDH, LWE, LPN) [QRW19, CH19, KNYY19a, KNYY19b, LQR+19].

Several recent works have also explored other relaxations of the standard notion of publicly-verifiable NIZKs such as the reusable designated-prover model (where there is a secret proving key and a public verification key) [KW18, KNYY19a] or the reusable preprocessing model (where both

[1] In [QRW19], they require the stronger notion where the CRS is a *uniformly random string*. In some of our constructions in this work, the CRS will be a *structured* string. We believe that this model is still meaningful as the CRS just needs to be sampled once and can be reused by arbitrarily many verifiers, and zero-knowledge holds as long as the CRS is properly sampled.

the proving and verifications keys are secret) [BCGI18, BCG+19]. In this work, our focus is on reusable designated-verifier NIZKs and publicly-verifiable NIZKs.

Dual-mode NIZKs. An appealing feature of several existing NIZK constructions [GOS06, GOS12, PS19] is they satisfy a "dual-mode" property. Namely, the CRS in these schemes can be sampled from one of two computationally indistinguishable distributions. One distribution yields *computational NIZK proofs* while the other yields *statistical NIZK arguments*. Dual-mode NIZKs are powerful primitives and a recent work has also studied generic constructions from obfuscation [HU19]. Most of the constructions we develop in this work naturally satisfy this dual-mode property.

1.1 Our Results

In this work, we develop new techniques for constructing statistical NIZKs for general NP languages that yield new constructions in both the reusable designated-verifier model and the standard CRS model. Our techniques enable the following new constructions:

- Under the k-Lin assumption in a *pairing-free* group (for any $k \geq 1$; recall that 1-$\mathsf{Lin} \equiv \mathsf{DDH}$), we obtain a statistical MDV-NIZK argument in the common *random* string model and a computational MDV-NIZK proof in the common *reference* string model.[2] This is the first construction of a statistical DV-NIZK argument (even ignoring malicious security) in a pairing-free group, and the first construction of a computational MDV-NIZK proof from a *static* assumption. Previously, computational MDV-NIZK proofs were only known from the interactive "one-more CDH" assumption [QRW19].

- Under the k-Lin assumption in $\mathbb{G}_1$ and the k-KerLin assumption in $\mathbb{G}_2$ of a pairing group (for any $k \geq 1$), we obtain a *publicly-verifiable* statistical NIZK argument in the common reference string model. Notably, the k-KerLin assumption is a *search* assumption that is implied by the standard k-Lin assumption [MRV15, KW15]. This is a *qualitatively weaker* assumption than existing pairing-based constructions of statistical NIZK arguments which rely on a *decisional* assumption (k-Lin) in *both* $\mathbb{G}_1$ and $\mathbb{G}_2$ [GOS06, GOS12].

- Under the QR assumption, we obtain a dual-mode MDV-NIZK in the common reference string model. Previously, we could only construct (publicly-verifiable) *computational* NIZKs from the QR assumption [BFM88] (or more generally, from factoring [FLS90, FLS99]), but nothing was known for statistical NIZKs or DV-NIZKs from these assumptions.

- Under the DCR assumption, we obtain a dual-mode MDV-NIZK in the common reference string model. This matches the recent construction described in [CDI+19], which realizes the result through a different approach (via reusable non-interactive secure computation).

We provide a detailed comparison of our constructions with existing NIZK constructions (in both the designated-verifier and the publicly-verifiable models) in Table 1. We describe the formal instantiations in Section 6.

[2] This is in fact a dual-mode NIZK, where one of the CRS distributions corresponds to the *uniform* distribution.

Construction	Model	Soundness	ZK	Assumption
[BFM88]	public	stat.	comp.	QR
[FLS90, FLS99]	public	stat.	comp.	trapdoor permutation
[SW14]	public	comp.	perf.	$i\mathcal{O}$ + one-way function
[CHK03]*	public	stat.	comp.	CDH ($\mathbb{G}_2$)
[GOS06, GOS12]*	public	perf./comp.	comp./perf.	k-Lin ($\mathbb{G}_1, \mathbb{G}_2$)
This work*	**public**	**comp.**	**stat.**	**k-Lin ($\mathbb{G}_1$), k-KerLin ($\mathbb{G}_2$)†**
[PS19]	public	stat./comp.	comp./stat.	LWE
[QRW19, CH19, KNYY19a]	DV	stat.	comp.	CDH
[QRW19]	MDV	stat.	comp.	one-more CDH
[LQR+19]	MDV	comp.	comp.	CDH/LWE/LPN
[CDI+19]	MDV	stat./comp.	comp./stat.	DCR
This work	**MDV**	**stat./comp.**	**comp./stat.**	**k-Lin‡/QR/DCR**

*This is a *pairing-based* construction. In the assumption column, we enumerate all of the necessary hardness assumptions to instantiate the scheme (in an asymmetric setting).

†The k-KerLin refers to the kernel k-Lin assumption [MRV15, KW15], which can be viewed as the *search* analog of the classic k-Lin assumption [BBS04, HK07, Sha07].

‡This is over a *pairing-free* group. The special case where $k = 1$ corresponds to the standard DDH assumption. In addition, if we consider the vanilla DV-NIZK model (without malicious security), there is a simple instantiation (over elliptic-curve groups) that achieves *perfect* zero-knowledge (Remark C.8).

Table 1: Comparison of our construction to existing multi-theorem NIZKs. We write "public" to denote the standard CRS model (with public proving and public verification), "DV" to denote the designated-verifier model, and "MDV" to denote the malicious-designated-verifier model. For soundness and zero-knowledge, we write "comp." to denote the computational variant of the property, "stat." to denote the statistical variant, and "perf." to denote the perfect variant. When a scheme supports a dual-mode CRS, we indicate the two modes by writing "stat./comp." For the pairing-based constructions, we list the necessary assumptions needed within each of the base groups $\mathbb{G}_1$ and $\mathbb{G}_2$ (assuming an asymmetric pairing).

From FLS to statistical NIZKs. All of our constructions follow the classic paradigm of Feige, Lapidot, and Shamir (FLS) [FLS99] who provide a general compiler from a NIZK in an idealized model (i.e., the "hidden-bits" model) to a computational NIZK proof in the CRS model. To date, all existing instantiations of the [FLS99] paradigm have yielded *computational NIZK proofs* in either the CRS model [FLS90, BY92, FLS99, CHK03, Gro10, Gol11, GR13, CL18] or the designated-verifier model [QRW19, CH19, KNYY19a]. In this work, we show how to adapt the general FLS paradigm to obtain new constructions of *statistical NIZK arguments* and more generally, *dual-mode NIZKs*. We provide a general overview of our techniques in Section 1.2.

We further note that previous statistical NIZK arguments from pairings, LWE, and DCR follow very different approaches. Our work can also be viewed as providing a *unified* approach to realizing these existing results—both computational and statistical, with the sole exception of the LWE-based scheme—via the FLS paradigm, while also improving upon some of these prior results, and obtaining new ones.

1.2 Technical Overview

We begin with a brief overview of the Feige-Lapidot-Shamir (FLS) framework [FLS90, FLS99] for constructing NIZK proofs for NP. We then describe how to adapt the main ideas from the FLS framework to obtain new constructions of (malicious) designated-verifier dual-mode NIZKs as well as publicly-verifiable statistical NIZK arguments.

The FLS framework. The starting point of the FLS construction is a NIZK in an idealized model called the "hidden-bits model." In this model, a trusted party generates a string of uniformly random bits $r_1, \ldots, r_\rho \in \{0,1\}$ and gives them to the prover. The prover then outputs a proof π along with a set of indices $I \subseteq [\rho]$. The verifier receives $(\pi, \{r_i\}_{i \in I})$ from the trusted party. The model guarantees that the prover cannot influence the value of any of the r_i's and the verifier does not learn anything about r_i for indices $i \notin I$. Feige et al. [FLS99] showed how to construct a NIZK with statistical soundness and perfect zero-knowledge in the hidden-bits model by adapting Blum's Σ-protocol for graph Hamiltonicity [Blu86]. Next, the FLS construction compiles a NIZK in the hidden-bits model into one in the CRS model by using the CRS to define the sequence of hidden bits. We recall the FLS compiler based on trapdoor permutations:

- The CRS contains the description of a *family* of trapdoor permutations over $\{0,1\}^\lambda$ together with ρ random strings $w_1, \ldots, w_\rho \in \{0,1\}^\lambda$ that are used to define a string of ρ hidden bits.
- A hidden-bits string is defined by sampling a permutation σ from the family of trapdoor permutations specified by the CRS, along with a trapdoor for computing σ^{-1}. In conjunction with w_i in the CRS, the permutation σ defines a hidden bit $r_i := \mathsf{hc}(\sigma^{-1}(w_i))$, where $\mathsf{hc}(\cdot)$ is a hard-core bit of σ. We refer to σ as a "commitment" to the hidden-bits string $r \in \{0,1\}^\rho$.
- The prover can open a commitment σ to a bit r_i by sending (i, r_i, u_i) where $u_i := \sigma^{-1}(w_i)$. The verifier checks that $\sigma(u_i) = w_i$ and that $\mathsf{hc}(u_i) = r_i$.

The security argument proceeds roughly as follows:

- Since hc is a hard-core bit, the value of any unopened bit r_i is *computationally hidden* given σ and w_i. The resulting NIZK satisfies computational zero-knowledge.
- The permutation σ and the string w_i *statistically determine* r_i, and the prover cannot open r_i to any value other than $\mathsf{hc}(\sigma^{-1}(w_i))$. The resulting NIZK satisfies statistical soundness. Note that a cheating prover can bias the bit r_i due to the adaptive choice of σ. The FLS construction works around this by leveraging the fact that if the commitment σ has length ℓ, then a malicious prover can bias at most ℓ of the ρ bits, and soundness holds as long as $\ell \ll \rho$.

Our approach. In this work, we start by showing how to realize a *dual-mode* variant of the hidden-bits model in the *designated-verifier* setting where the underlying commitment to the random bits is either statistically binding or statistically hiding. This "dual-mode" property yields either a *computational DV-NIZK proof* or a *statistical DV-NIZK argument* depending on how the CRS is sampled (similar to previous dual-mode NIZKs [GOS06, GOS12, PS19]). We then show how to extend one of our constructions to the *publicly-verifiable* setting.

An instantiation from DDH. We first sketch our construction from the DDH assumption. Here, we will work with a (multiplicative) group $\mathbb{G}$ of prime order p and generator g. For a vector $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_p^n$, we write $g^{\mathbf{v}}$ to denote a vector of group elements $(g^{v_1}, \ldots, g^{v_n})$. Analogous to the FLS construction from trapdoor permutations, the CRS contains

- the description $g^{\mathbf{v}}$ of a function, where $\mathbf{v} \xleftarrow{R} \mathbb{Z}_p^{\rho+1}$ and $g^{\mathbf{v}}$ plays a role similar to the *family* of trapdoor permutations in the FLS construction;
- $g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho}$ where each $\mathbf{w}_i \in \mathbb{Z}_p^{\rho+1}$ plays a role similar to $w_i \in \{0,1\}^\lambda$.

In our construction, we will vary the distribution of $\mathbf{w}_i$ (but *not* $\mathbf{v}$) as follows:

- If we want *statistically-binding* "hidden bits," then we sample $\mathbf{w}_i \leftarrow s_i \mathbf{v}$, where $s_i \xleftarrow{R} \mathbb{Z}_p$.
- If we want *statistically-hiding* "hidden bits," then we sample $\mathbf{w}_i \xleftarrow{R} \mathbb{Z}_p^{\rho+1}$.

Thanks to the DDH assumption, $(g^{\mathbf{v}}, g^{s_i \mathbf{v}})$ is pseudorandom, and therefore, these two CRS distributions are computationally indistinguishable.[3] As with the construction from trapdoor permutations, the hidden bit r_i is a function of the CRS components $g^{\mathbf{v}}, g^{\mathbf{w}_i}$ together with an additional message σ from the prover. Concretely, the prover samples a random $\mathbf{y} \xleftarrow{R} \mathbb{Z}_p^{\rho+1}$ and sends $\sigma = g^{\mathbf{y}^\mathsf{T}\mathbf{v}} \in \mathbb{G}$. In conjunction with $g^{\mathbf{w}_i}$ in the CRS, the vector $\mathbf{y}$ defines a hidden bit $r_i := H(g^{\mathbf{y}^\mathsf{T}\mathbf{w}_i})$, where $H \colon \mathbb{G} \to \{0,1\}$ is a universal hash function. Importantly, while the description $g^{\mathbf{v}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho}$ in the CRS grows with ρ, the prover's message σ does not. Now, observe that:

- In binding mode where $\mathbf{w}_i = s_i \mathbf{v}$, we have $\mathbf{y}^\mathsf{T}\mathbf{w}_i = s_i \mathbf{y}^\mathsf{T}\mathbf{v}$. Then, $r_i = H(g^{\mathbf{y}^\mathsf{T}\mathbf{w}_i}) = H(g^{s_i \mathbf{y}^\mathsf{T}\mathbf{v}}) = H(\sigma^{s_i})$ is fully determined by the commitment $\sigma = g^{\mathbf{y}^\mathsf{T}\mathbf{v}}$ together with $g^{\mathbf{v}}, g^{\mathbf{w}_i}$ in the CRS.
- In hiding mode where $\mathbf{w}_i \xleftarrow{R} \mathbb{Z}_p^{\rho+1}$, the quantity $g^{\mathbf{y}^\mathsf{T}\mathbf{w}_i}$ is completely hidden given $g^{\mathbf{y}^\mathsf{T}\mathbf{v}}$ along with $g^{\mathbf{v}}, g^{\mathbf{w}_i}$ in the CRS, provided that $\mathbf{v}$ and $\mathbf{w}_i$ are linearly independent. More generally, perfect hiding holds as long as the vectors $\mathbf{v}, \mathbf{w}_1, \ldots, \mathbf{w}_\rho$ are linearly independent over $\mathbb{Z}_p^{\rho+1}$.

Next, to open the bit r_i, the prover will send along $g^{\mathbf{y}^\mathsf{T}\mathbf{w}_i}$. To ensure that a cheating prover computes this quantity correctly in the *designated-verifier* model, we rely on techniques using the Cramer-Shoup hash-proof system [CS98, CS02, CKS08] (and also used to construct computational DV-NIZK proofs from CDH [QRW19, CH19, KNYY19a]):

- The verifier's public key consists of components $g^{\mathbf{z}_i} := g^{a\mathbf{w}_i + b_i \mathbf{v}}$ where $a, b_i \xleftarrow{R} \mathbb{Z}_p$ are secret coefficients chosen by the verifier. The *secret* verification key is the scalars $(a, b_1, \ldots, b_\rho)$.
- The prover sends $g^{u_i} := g^{\mathbf{y}^\mathsf{T}\mathbf{z}_i} \in \mathbb{G}$ in addition to $\sigma = g^c := g^{\mathbf{y}^\mathsf{T}\mathbf{v}} \in \mathbb{G}$ and $g^{t_i} := g^{\mathbf{y}^\mathsf{T}\mathbf{w}_i} \in \mathbb{G}$.
- The verifier checks that $g^{u_i} = (g^{t_i})^a (g^c)^{b_i}$ using (a, b_i).

In the statistically-binding mode where $\mathbf{w}_i = s_i \mathbf{v}$, we have $\mathbf{z}_i = (a s_i + b_i)\mathbf{v}$, so (a, b_i) has (statistical) entropy given $\mathbf{v}, \mathbf{w}_i, \mathbf{z}_i$. Roughly speaking, reusable soundness then follows from the analysis of the Cramer-Shoup CCA-secure encryption scheme [CS98, CS02, CKS08] to enforce the consistency check $t_i = s_i c$. In conjunction with a NIZK in the hidden-bits model, we thus obtain a dual-mode

[3] This idea of encoding either a full-rank matrix in the exponent or a rank-1 matrix in the exponent also featured in the construction of lossy public-key encryption from the Matrix Diffie-Hellman assumptions [HJR16].

DV-NIZK from the DDH assumption. This construction generalizes very naturally to the k-Lin family of assumptions [BBS04, HK07, Sha07, EHK+13] for any $k \geq 1$ (where in particular, 1-Lin is the DDH assumption). Concretely, we make the following substitutions to the above construction:

$$\begin{aligned} \mathbf{v} \in \mathbb{Z}_p^{\rho+1} &\mapsto \mathbf{V} \in \mathbb{Z}_p^{(\rho+1)\times k} \\ s_i, b_i \in \mathbb{Z}_p &\mapsto \mathbf{s}_i, \mathbf{b}_i \in \mathbb{Z}_p^k \\ t_i, u_i, c \in \mathbb{Z}_p &\mapsto \mathbf{t}_i, \mathbf{u}_i, \mathbf{c} \in \mathbb{Z}_p^k \end{aligned}$$

We provide the full details and security analysis in Section 4.1.

Extending to QR/DCR. Our DDH construction readily generalizes to the subgroup indistinguishability family of assumptions [BG10] (which generalize the QR [GM82] and DCR [Pai99] assumptions). While there are some technical differences in our concrete instantiations from QR (Section 5) and DCR (Appendix E), all of the main ideas can be described via the conceptually-simpler language of subgroup indistinguishability. This is the approach we take in this overview, and we refer to the technical sections for the full details. First, the subgroup indistinguishability assumption says that the distributions (g, h, g^{r_1}) and $(g, h, g^{r_1} h^{r_2})$ are computationally indistinguishable, where g, h generate subgroups of co-prime order m_g, m_h, respectively, and $r_1 \xleftarrow{R} \mathbb{Z}_{m_g}$, $r_2 \xleftarrow{R} \mathbb{Z}_{m_h}$.

Similar to the DDH instantiation, the CRS contains a function $g^{\mathbf{v}}$ (where $\mathbf{v} \xleftarrow{R} \mathbb{Z}_{m_g m_h}^{\rho}$) together with additional components $g^{s_1 \mathbf{v}} h^{\hat{\mathbf{w}}_1}, \ldots, g^{s_\rho \mathbf{v}} h^{\hat{\mathbf{w}}_\rho}$, where $\hat{\mathbf{w}}_i = \mathbf{0}$ in binding mode and $\hat{\mathbf{w}}_i = \mathbf{e}_i$ in hiding mode. Here $\mathbf{e}_i$ is the basis vector whose i^{th} index is 1. Under the subgroup indistinguishability assumption, these two distributions are computationally indistinguishable.

Next, the hidden bit r_i is a function of the CRS components $g^{\mathbf{v}}$ and $g^{s_i \mathbf{v}} h^{\hat{\mathbf{w}}_i}$ together with an additional commitment σ from the prover. Specifically, the prover samples a vector $\mathbf{y} = (y_1, \ldots, y_\rho) \xleftarrow{R} \mathbb{Z}_{m_g m_h}^{\rho}$ and computes

$$\sigma := g^{\mathbf{y}^\mathsf{T} \mathbf{v}} \quad \text{and} \quad t_i := g^{s_i \mathbf{y}^\mathsf{T} \mathbf{v}} h^{\mathbf{y}^\mathsf{T} \hat{\mathbf{w}}_i} \quad \text{and} \quad r_i := H(t_i), \tag{1.1}$$

where H is a hash function. Now, observe that:

- In binding mode where $\hat{\mathbf{w}}_i = \mathbf{0}$, then $t_i = g^{s_i \mathbf{y}^\mathsf{T} \mathbf{v}} = \sigma^{s_i}$. Thus, t_i (and correspondingly, r_i) is fully determined by the commitment σ and the components $g^{\mathbf{v}}$, $g^{s_i \mathbf{v}} h^{\hat{\mathbf{w}}_i} = g^{s_i \mathbf{v}}$ in the CRS.
- In hiding mode where $\hat{\mathbf{w}}_i = \mathbf{e}_i$, then $t_i = g^{s_i \mathbf{w}^\mathsf{T} \mathbf{y}} h^{y_i}$. Since g and h generate subgroups of co-prime order m_g and m_h, respectively, we can appeal to the Chinese remainder theorem to argue that the commitment $\sigma = g^{\mathbf{y}^\mathsf{T} \mathbf{v}}$ *perfectly* hides the value of $\mathbf{y}$ mod m_h. Since $\mathbf{y}$ is uniform over $\mathbb{Z}_{m_g m_h}$, this means that $t_1, \ldots, t_i$ have at least $\log m_h$ bits of statistical entropy given σ (and the components of the CRS).

  In the DCR construction, $m_h = N$ is a product of two large primes, so we can use a standard universal hash function to extract a uniformly random bit [HILL99].

  In the QR construction, $m_h = 2$, so each component t_i contains just one bit of entropy, and we cannot appeal to the leftover hash lemma. In this case, we adapt an idea from [DGI+19] (for constructing trapdoor hash functions from QR) and use a deterministic function to extract the bit from t_i. We provide the full details in Section 5.

Finally, to open a bit r_i, the prover provides σ, t_i, along with a proof that t_i and σ are consistent (i.e., there exists some $\mathbf{y}$ such that Eq. (1.1) hold). Here, we use the same techniques as in the DDH setting (i.e., using the Cramer-Shoup hash-proof system) to implement this. In the QR setting, we encounter some challenges because the order of the subgroup generated by h is polynomial-sized, which allows the adversary to break soundness with noticeable probability. To amplify soundness, we essentially embed multiple copies of the Cramer-Shoup hash-proof system and ensure that the proof verifies only if *all* copies verify (while retaining *reusable* soundness). We refer to Section 5 and Appendix E for the full analysis of the QR and DCR constructions, respectively.

Handling malicious verifiers. All of the constructions described thus far are zero-knowledge only if the verifier samples its public verification key *honestly*. However, if the verifier can choose its key *arbitrarily*, then it can break zero-knowledge. To see this, consider again the DDH construction (in hiding mode). There, the CRS contains elements $g^{\mathbf{v}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho}$, and a verifier's public key is $(g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_\rho})$ where $\mathbf{z}_i = a\mathbf{w}_i + b_i\mathbf{v}$. To generate a hidden-bits string r, the prover samples $\mathbf{y} \xleftarrow{R} \mathbb{Z}_p^{\rho+1}$ and sets $r_i = H(g^{\mathbf{y}^\mathsf{T}\mathbf{w}_i})$. To open a bit r_i, the prover computes $g^{t_i} = g^{\mathbf{y}^\mathsf{T}\mathbf{w}_i}$ and $g^{u_i} = g^{\mathbf{y}^\mathsf{T}\mathbf{z}_i}$. In order to appeal to security of the underlying NIZK in the hidden-bits model, we require that the commitment $\sigma = g^{\mathbf{y}^\mathsf{T}\mathbf{v}}$, the value of r_i, and the opening (g^{t_i}, g^{u_i}) do not leak information about any other (unopened) bit r_j. This is the case when all of the verification key components $\mathbf{z}_i$ are generated honestly. In this case, $\mathbf{v}, \mathbf{w}_1, \ldots, \mathbf{w}_\rho$ are linearly independent, and $\mathbf{z}_i$ is a function of only $\mathbf{v}$ and $\mathbf{w}_i$. However, a malicious verifier can choose $\mathbf{z}_i = \mathbf{w}_j$ for some $j \neq i$. Then, if the honest prover computes an opening to r_i, it will also compute $g^{u_i} = g^{\mathbf{y}^\mathsf{T}\mathbf{z}_i} = g^{\mathbf{y}^\mathsf{T}\mathbf{w}_j}$, which completely leaks the value of r_j. As such, the basic scheme is insecure against a malicious verifier.

This problem where an opening to r_i can leak information about the value r_j for $j \neq i$ is the same problem encountered in the basic DV-NIZK from [QRW19]. In this work, we adopt the same general strategy as them to defend against malicious verifiers. At a high-level, the approach of [QRW19] for achieving security against malicious verifiers is to use the basic scheme above to generate a hidden-bits string $r'_1, \ldots, r'_\ell$ of length $\ell \gg \rho$. Each of the ρ hidden bits $r_1, \ldots, r_\rho$ is then derived as a sparse pseudorandom combination of the bits $r'_1, \ldots, r'_\ell$. More specifically, the prover chooses a mapping φ that maps each index $i \in [\rho]$ onto a set $\varphi(i) \subseteq [\ell]$. Each bit r_i is a deterministic function of r'_j for $j \in \varphi(i)$. To open a bit r_i, the prover instead opens up all bits r'_j for $j \in \varphi(i)$. The length ℓ and the size $|\varphi(i)|$ of the sets are chosen so as to ensure that for all unopened bits $j \in [\rho]$, there is at least one index $k \in \varphi(j)$ such that r'_k is hidden from the verifier, which ideally, is sufficient to mask the value of r_j. Quach et al. show how to implement this idea by relying on a one-more CDH assumption (in conjunction with somewhere equivocal PRFs [HJO+16]), and a complex rewinding argument in the security proof. In our setting, the algebraic structure of our construction enables us to make a conceptually-simpler *information-theoretic* argument (and only needing to assume a PRG). As such, we are able to obtain a *dual-mode* MDV-NIZK from the DDH (and more generally, k-Lin)as well as the QR and DCR assumptions.

We give a brief overview of how we extend the basic DDH construction sketched above to achieve security against malicious verifiers. The same idea extends to the QR and DCR constructions. Specifically, we use our basic construction to generate a hidden-bits string of length $\ell \gg \rho$ as follows:

- The CRS (in hiding mode) consists of group elements $g^{\mathbf{v}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\ell}$, where $\mathbf{v}, \mathbf{w}_1, \ldots, \mathbf{w}_\ell \xleftarrow{R} \mathbb{Z}_p^{\ell+1}$. With overwhelming probability, these vectors are linearly independent.
- The honest verifier's public key is $(g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_\ell})$, constructed in the usual manner.

- The prover's commitment is a vector $\mathbf{y} \in \mathbb{Z}_p^{\ell+1}$ as well as a seed s for a PRG.[4] The PRG outputs a collection of ρ blocks, where each block consists of a set $S_i \subseteq [\ell]$ and a vector $\boldsymbol{\alpha} \in \mathbb{Z}_p^{\ell}$. The hidden bit r_i is determined by first computing $g^{t_j} = g^{\mathbf{y}^\mathsf{T}\mathbf{w}_j}$ for all $j \in S_i$ and defining $r_i := H(\prod_{j \in S_i} g^{\alpha_j t_j})$.
- The opening for r_i consists of $g^{t_j} = g^{\mathbf{y}^\mathsf{T}\mathbf{w}_j}$ and $g^{u_j} = g^{\mathbf{y}^\mathsf{T}\mathbf{z}_j}$ for all $j \in S_i$.

Our goal is to show that even for an adversarially-chosen verification key, the commitment σ and the opening $(\{g^{t_j}, g^{u_j}\}_{j \in S_i})$ to a bit r_i does not leak any information about r_j whenever $j \neq i$.[5] By construction, the opening to r_i is determined by $\mathbf{y}^\mathsf{T}\mathbf{v}$, $\mathbf{y}^\mathsf{T}\mathbf{w}_j$, and $\mathbf{y}^\mathsf{T}\mathbf{z}_j$ for $j \in S_i$ (where the set S_i is *pseudorandom*). Take any index $i^* \neq i$. Then, if there exists $j^* \in \varphi(i^*)$ such that $\mathbf{w}_{j^*}$ is linearly independent of $\{\mathbf{v}, \mathbf{w}_j, \mathbf{z}_j\}_{j \in S_i}$, then the value of $\mathbf{y}^\mathsf{T}\mathbf{w}_{j^*}$ is independent and uniformly random given the view of the adversary (since the honest prover samples $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell+1}$). In this case, the value $g^{t_{j^*}} = g^{\mathbf{y}^\mathsf{T}\mathbf{w}_{j^*}}$ remains uniformly random and statistically hides r_{i^*}. Thus, it suffices to set ℓ and $|S_i|$ so that there will always exist $j^* \in \varphi(i^*)$ where $\mathbf{w}_{j^*}$ is linearly independent of $\{\mathbf{v}, \mathbf{w}_j, \mathbf{z}_j\}_{j \in S_i}$ with overwhelming probability. In the case of our DDH construction, we can set $|S_i| = \lambda$, where λ is a security parameter, and $\ell = 3\rho^2\lambda$ to satisfy this property. We provide the full analysis of our DDH (more generally, its generalization to the k-Lin assumption) in Section 4.3, our QR construction in Section 5.2 and our DCR construction in Appendix E.2.

Public verifiability via pairings. All of the constructions we have described so far operate in the designated-verifier model because our constructions rely on a Cramer-Shoup-style hash proof system to argue consistency between a commitment and the opening. If we can instead *publicly* check consistency between a commitment and its opening, then the resulting scheme becomes publicly verifiable. For the DDH construction, we can implement the consistency check using a pairing (this is the approach taken in [CHK03] to obtain a computational NIZK proof). In this work, we develop a similar approach to obtain a statistical NIZK argument from pairings.

In particular, let $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be an (asymmetric) pairing. Let g_1, g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. At a high level, we implement the DDH scheme in $\mathbb{G}_1$ and use $\mathbb{G}_2$ for verification. More specifically, the CRS is $g_1^{\mathbf{v}}, g_1^{\mathbf{w}_1}, \ldots, g_1^{\mathbf{w}_\rho}$, and the verification key is $g_1^{(a\mathbf{w}_1 + b_1\mathbf{v})}, \ldots, g_1^{(a\mathbf{w}_\rho + b_\rho\mathbf{v})}$. The commitment, hidden-bits sequence, and openings are defined as before:

$$\sigma = g_1^c = g_1^{\mathbf{y}^\mathsf{T}\mathbf{v}} \quad , \quad r_i = H(g_1^{\mathbf{y}^\mathsf{T}\mathbf{w}_i}) \quad , \quad g_1^{t_i} = g_1^{\mathbf{y}^\mathsf{T}\mathbf{w}_i} \quad \text{and} \quad g_1^{u_i} = g_1^{\mathbf{y}^\mathsf{T}(a\mathbf{w}_i + b_i\mathbf{v})}.$$

In the designated-verifier setting, the verifier checks $g_1^{u_i} \stackrel{?}{=} (g_1^{t_i})^a (g_1^c)^{b_i}$. A direct approach for public verification is to include $g_2^a, g_2^{b_1}, \ldots, g_2^{b_\rho}$ as part of the verification key, and check the following:

$$e(g_1^{u_i}, g_2) \stackrel{?}{=} e(g_1^{t_i}, g_2^a) \cdot e(g_1^c, g_2^{b_i}).$$

While this approach is *correct*, it is unclear to argue soundness (even against computationally-bounded adversaries). In the designated-verifier setting, the soundness analysis critically relies on

[4] We require a PRG because the prover's message needs to be *succinct* in order to argue soundness of the resulting NIZK in the FLS paradigm. Thus, we rely on a PRG for compression. Note that even though we rely on a computational assumption, we can still show *statistical* zero-knowledge. The security proof only requires that there are no efficient statistical tests that can distinguish the output of the PRG from a random string (which is implied by PRG security).

[5] To show adaptive, multi-theorem zero-knowledge, we in fact show an even stronger *simulation* property. We refer to Section 3 for more details.

the verification coefficients a, b_i being hidden from the adversary, and it is unclear how to make such an argument when the adversary is given $g_2^a, g_2^{b_i}$.

To base hardness on a concrete cryptographic assumption, we leverage a technique from [KW15], who describe a general method to "securely publish" the verification key in the exponent (as we hoped to do in our initial attempt above) with a concrete security reduction to a *search* assumption in $\mathbb{G}_2$. This yields a general compiler from a designated-verifier scheme with unconditional soundness to a publicly-verifiable scheme with computational soundness, at the expense of requiring a pairing and a *search* assumption in $\mathbb{G}_2$. The compiler preserves zero-knowledge of the underlying scheme.

Concretely, instead of scalar verification coefficients a, b_i, we instead sample vectors $\mathbf{a}, \mathbf{b}_i \xleftarrow{R} \mathbb{Z}_p^2$, and publish $g_1^{\mathbf{w}_i\mathbf{a}^\top + \mathbf{v}\mathbf{b}_i^\top}$ for each $i \in [\rho]$ in the CRS. The public verification components will consist of $g_2^{\mathbf{d}}, g_2^{\mathbf{a}^\top\mathbf{d}}, g_2^{\mathbf{b}_1^\top\mathbf{d}}, \ldots, g_2^{\mathbf{b}_\rho^\top\mathbf{d}}$, where $\mathbf{d} \in \mathbb{Z}_p^2$. The key observation is that $\mathbf{a}, \mathbf{b}_1, \ldots, \mathbf{b}_\rho$ have *statistical entropy* even given the public components $g_2^{\mathbf{d}}, g_2^{\mathbf{a}^\top\mathbf{d}}, g_2^{\mathbf{b}_1^\top\mathbf{d}}, \ldots, g_2^{\mathbf{b}_\rho^\top\mathbf{d}}$. The commitment, hidden-bits sequence, and openings are still computed as before, except the verification component $g_1^{u_i}$ is replaced with $g_1^{\mathbf{u}_i^\top} = g_1^{\mathbf{y}^\top(\mathbf{w}_i\mathbf{a}^\top + \mathbf{v}\mathbf{b}_i^\top)}$. The verification relation now checks

$$e(g_1^{\mathbf{u}_i^\top}, g_2^{\mathbf{d}}) \stackrel{?}{=} e(g_1^{t_i}, g_2^{\mathbf{a}^\top\mathbf{d}}) \cdot e(g_1^{c}, g_2^{\mathbf{b}_i^\top\mathbf{d}}).$$

Since the verification coefficients $\mathbf{a}, \mathbf{b}_1, \ldots, \mathbf{b}_\rho$ have statistical entropy given the public key, we can appeal to DDH in $\mathbb{G}_1$ and the 1-KerLin assumption (a *search* assumption that is *weaker* than DDH) over $\mathbb{G}_2$ to argue soundness of the resulting construction. This yields a publicly-verifiable statistical NIZK argument in the common *reference* string model. We provide the full description and analysis (generalized to the k-Lin and k-KerLin family of assumptions for any $k \geq 1$) in Section 4.2.

Our pairing-based construction does not appear to have a dual mode and it is unclear how to modify this construction to obtain computational NIZK proofs. We do note that computational NIZK proofs can be built directly from pairings (under the CDH assumption in $\mathbb{G}_1$) also by following the FLS paradigm [CHK03]. At the same time, it is also unclear how to adapt the [CHK03] construction to obtain statistical NIZK arguments.

A unifying abstraction: dual-mode hidden-bits generators. We unify the different algebraic constructions through the abstraction of a "dual-mode hidden-bits generator." Previously, Quach et al. [QRW19] introduced the notion of a *hidden-bits generator* (HBG) and showed how to use an HBG to implement the classic FLS paradigm in both the designated-verifier and the publicly-verifiable settings. Very briefly, an HBG with output size ρ consists of four main algorithms (Setup, KeyGen, GenBits, Verify):

- The Setup algorithm outputs a common reference string crs, and KeyGen generates a public key pk along with a (possibly secret) verification key sk.
- The GenBits algorithm outputs a short commitment σ together with a sequence of hidden bits $r \in \{0,1\}^\rho$ as well as openings $\{\pi_i\}_{i\in[\rho]}$.
- The Verify algorithm takes an index $i \in [\rho]$, a bit $r_i \in \{0,1\}$, and an opening π_i and either accepts or rejects the proof.

The main security requirements are *statistical binding* (i.e., no adversary can produce a commitment σ and valid openings π_i, π_i' that open to 0 and 1 for the same index) and *computational hiding*

(i.e., an honestly-generated commitment σ and set of openings $\{r_i, \pi_i\}_{i \in I}$ should hide all unopened bits r_j for $j \notin I$ from any computationally-bounded adversary). Quach et al. show that an HBG with these properties can be combined directly with a NIZK in the hidden-bits model to obtain a computational NIZK proof in the CRS model. If the HBG is in the (malicious) designated-verifier model, then so is the resulting NIZK.

In this work, we extend this framework by introducing the notion of a dual-mode HBG where the CRS can be generated in one of two modes: a *binding* mode where the HBG satisfies statistical binding (as in [QRW19]) and a *hiding* mode where the HBG satisfies a stronger notion of *statistical hiding* (i.e., the unopened bits are statistically hidden given the CRS, the commitment σ and any subset of opened bits $\{(r_i, \pi_i)\}_{i \in I}$). In our case, we impose an even stronger equivocation property in the hiding mode: namely, given any any set of indices $I \subseteq [\rho]$ and any assignment $r_I \in \{0,1\}^{|I|}$ to that set, it is possible to simulate a commitment σ and a set of openings $\{\pi_i\}_{i \in I}$ that is statistically indistinguishable from the output of the honest generator. This allows us to directly argue *adaptive* and *multi-theorem*[6] statistical zero-knowledge for the resulting NIZK construction. We give our formal definition in Section 3, and describe our construction of dual-mode (designated-verifier) NIZKs from dual-mode (designated-verifier) HBGs in Section 3.1. In Section 4 and Section 5 and Appendix E, we show how to construct dual-mode HBGs from the k-Lin, QR, and DCR assumptions.

2 Preliminaries

Throughout this work, we write λ (oftentimes implicitly) to denote the security parameter. For a positive integer $n \in \mathbb{N}$, we write $[n]$ to denote the set $\{1, \ldots, n\}$. We will typically use bold lowercase letters (e.g., $\mathbf{v}, \mathbf{w}$) to denote vectors and bold uppercase letters (e.g., $\mathbf{A}, \mathbf{B}$) to denote matrices. For a vector $\mathbf{v} \in \mathbb{Z}_p^n$, we will use non-boldface letters to refer to its components; namely, we write $\mathbf{v} = (v_1, \ldots, v_n)$. For a (sorted) set of indices $I = \{i_1, \ldots, i_m\} \subseteq [n]$, we write $\mathbf{v}_I$ to denote the sub-vector $(v_{i_1}, \ldots, v_{i_m})$. For a matrix $\mathbf{A}$, we write $\mathsf{span}(\mathbf{A})$ to denote the vector space spanned by the columns of $\mathbf{A}$.

We say that a function f is negligible in λ, denoted $\mathsf{negl}(\lambda)$, if $f(\lambda) = o(1/\lambda^c)$ for all $c \in \mathbb{N}$. We write $\mathsf{poly}(\lambda)$ to denote a function bounded by a fixed polynomial in λ. We say an event happens with negligible probability if the probability of the event happening is negligible, and that it happens with overwhelming probability if its complement occurs with negligible probability. We say that an algorithm is efficient if it runs in probabilistic polynomial-time in the length of its inputs. We say that two families of distributions $\mathcal{D}_1 = \{\mathcal{D}_{1,\lambda}\}_{\lambda \in \mathbb{N}}$ and $\mathcal{D}_2 = \{\mathcal{D}_{2,\lambda}\}_{\lambda \in \mathbb{N}}$ are computationally indistinguishable if no efficient adversary can distinguish samples from $\mathcal{D}_1$ and $\mathcal{D}_2$ except with negligible probability, and we denote this by writing $\mathcal{D}_1 \overset{c}{\approx} \mathcal{D}_2$. For two distributions $\mathcal{D}_1$, $\mathcal{D}_2$, we write $\Delta(\mathcal{D}_1, \mathcal{D}_2)$ to denote the statistical distance between $\mathcal{D}_1$ and $\mathcal{D}_2$. We write $\mathcal{D}_1 \overset{s}{\approx} \mathcal{D}_2$ to denote that $\mathcal{D}_1$ and $\mathcal{D}_2$ are statistically indistinguishable: namely, that $\Delta(\mathcal{D}_1, \mathcal{D}_2) = \mathsf{negl}(\lambda)$. For a finite set S, we write $x \xleftarrow{R} S$ to denote that x is sampled uniformly at random from S. For a distribution $\mathcal{D}$, we write $x \leftarrow \mathcal{D}$ to denote that x is sampled from $\mathcal{D}$. We now recall the definition of a pseudorandom generator (PRG).

[6] We can also use the transformation from [FLS99] to generically go from single-theorem zero-knowledge to multi-theorem zero-knowledge, but at the expense of making *non-black-box* use of a PRG. Our approach yields a direct construction of multi-theorem zero-knowledge without needing to make non-black-box use of cryptography. We discuss this in greater detail in Remark 2.12.

Definition 2.1 (Pseudorandom Generator)**.** A pseudorandom generator (PRG) with seed length $\kappa = \kappa(\lambda)$ and output length $\ell = \ell(\lambda)$ is an efficiently-computable function $G\colon \{0,1\}^\kappa \to \{0,1\}^\ell$ with the property that for all efficient adversaries $\mathcal{A}$,

$$\left|\Pr[\mathsf{s} \xleftarrow{R} \{0,1\}^\kappa : \mathcal{A}(G(\mathsf{s})) = 1] - \Pr[t \xleftarrow{R} \{0,1\}^\ell : \mathcal{A}(t) = 1]\right| = \mathsf{negl}(\lambda).$$

2.1 Hash Functions and Randomness Extraction

We now recall the leftover hash lemma [HILL99] and some properties of hash functions.

Definition 2.2 (Uniformity of Hash Functions)**.** A family of hash functions $\mathcal{H} = \{H\colon \mathcal{X} \to \mathcal{Y}\}$ satisfies statistical uniformity if

$$\{H \xleftarrow{R} \mathcal{H}, x \xleftarrow{R} \mathcal{X} : (H, H(x))\} \overset{s}{\approx} \{H \xleftarrow{R} \mathcal{H}, y \xleftarrow{R} \mathcal{Y} : (H, y)\}.$$

When the two distributions above are identically distributed, then $\mathcal{H}$ satisfies perfect uniformity.

Definition 2.3 (Universal Hash Function)**.** A family of hash functions $\mathcal{H} = \{H\colon \mathcal{X} \to \{0,1\}^\ell\}$ is universal if for any $x_1 \neq x_2 \in \mathcal{X}$, we have that $\Pr[H \xleftarrow{R} \mathcal{H} : H(x_1) = H(x_2)] \leq 1/2^\ell$.

Lemma 2.4 (Leftover Hash Lemma [HILL99])**.** *Let $\mathcal{H} = \{H\colon \mathcal{X} \to \{0,1\}^\ell\}$ be a universal family of hash functions. Take any $\varepsilon > 0$ and let $\mathcal{D}$ be a distribution over $\mathcal{X}$ with min-entropy $H_\infty(\mathcal{D}) \geq \ell + 2\log(1/\varepsilon)$. Then, $\Delta((H, H(x)), (H, U)) = \varepsilon$, where $H \xleftarrow{R} \mathcal{H}$, $x \leftarrow \mathcal{D}$, and $U \xleftarrow{R} \{0,1\}^\ell$.*

Corollary 2.5 (Universal Hash Functions are Statistically Uniform)**.** *Let $\mathcal{H} = \{H\colon \mathcal{X} \to \{0,1\}^\ell\}$ be a universal family of hash functions. If $|\mathcal{X}| \geq \ell + \omega(\log\lambda)$, then $\mathcal{H}$ satisfies statistical uniformity.*

We also need a variant of the leftover hash lemma that allows extracting randomness from sources that may be correlated with the seed. We use the following lemma from [YYHK16]:

Lemma 2.6 (Randomness Extraction from Seed-Dependent Sources [YYHK16])**.** *Let $\mathcal{H} = \{H\colon \mathcal{X} \to \{0,1\}^\ell\}$ be a family of pairwise-independent hash functions. Take any $\varepsilon > 0$ and let $\mathcal{D} = \{D_i : H_\infty(D_i) \geq \ell + 2\log(1/\varepsilon)\}_{i\in[M]}$ be a collection of distributions D_i over $\mathcal{X}$ where each D_i has min-entropy at least $\ell + 2\log(1/\varepsilon)$. Then, for all algorithms $\mathcal{A}$ that takes as input a hash function $H \in \mathcal{H}$ and outputs an index $i \in [M]$,*

$$\Delta((H, H(x)), (H, U)) \leq |\mathcal{D}|\,\varepsilon = M\varepsilon,$$

where $H \xleftarrow{R} \mathcal{H}$, $i \xleftarrow{R} \mathcal{A}(H)$, $x \leftarrow D_i$, and $U \xleftarrow{R} \{0,1\}^\ell$.

2.2 NIZKs in the Hidden-Bits Model

In this section, we recall the notion of a NIZK in the hidden-bits model [FLS99]. Our presentation is adapted from the description from [QRW19, CH19, KNYY19a].

Definition 2.7 (NIZKs in the Hidden-Bits Model)**.** Let $\mathcal{L} \subseteq \{0,1\}^n$ be an NP language associated with an NP relation $\mathcal{R}$ with $n = n(\lambda)$. A non-interactive zero-knowledge proof in the hidden-bits model for $\mathcal{L}$ consists of a tuple $\Pi_{\mathsf{HBM}} = (\mathsf{Prove}, \mathsf{Verify})$ and a parameter $\rho = \rho(\lambda, n)$ with the following properties:

- $\mathsf{Prove}(1^\lambda, r, x, w) \to (I, \pi)$: On input the security parameter λ, a string $r \in \{0,1\}^\rho$, a statement $x \in \{0,1\}^n$ and a witness w, this algorithm outputs a set of indices $I \subseteq [\rho]$ and a proof π.
- $\mathsf{Verify}(1^\lambda, I, r_I, x, \pi) \to \{0,1\}$: On input the security parameter λ, a subset $I \subseteq [\rho]$, a string $r_I \in \{0,1\}^{|I|}$, a statement $x \in \{0,1\}^n$ and a proof π, the verification algorithm outputs a bit $b \in \{0,1\}$.

Moreover, Π_{HBM} satisfies the following properties:

- **Completeness:** For all $(x, w) \in \mathcal{R}$ and $r \in \{0,1\}^\rho$,
$$\Pr[(I, \pi) \leftarrow \mathsf{Prove}(1^\lambda, r, x, w) : \mathsf{Verify}(1^\lambda, I, r_I, x, \pi) = 1] = 1.$$
- **Statistical soundness:** For all unbounded provers $\mathcal{P}^*$, we have that for $r \xleftarrow{R} \{0,1\}^\rho$ and $(x, \pi, I) \leftarrow \mathcal{P}^*(1^\lambda, r)$,
$$\Pr[x \notin \mathcal{L} \ \wedge \ \mathsf{Verify}(1^\lambda, I, r_I, x, \pi) = 1] = \mathsf{negl}(\lambda).$$
We will oftentimes refer to the above probability as the *soundness error*.
- **Perfect zero-knowledge:** There exists an efficient simulator $\mathcal{S}$ such that for all unbounded verifiers $\mathcal{V}^*$, if we take $(x, w) \leftarrow \mathcal{V}^*(1^\lambda)$, $r \xleftarrow{R} \{0,1\}^\rho$, $(I, \pi) \leftarrow \mathsf{Prove}(1^\lambda, r, x, w)$, and $(\widetilde{I}, \widetilde{r_I}, \widetilde{\pi}) \leftarrow \mathcal{S}(1^\lambda, x)$, and moreover if $\mathcal{R}(x, w) = 1$, then the following two distributions are identically distributed:
$$(I, r_I, \pi) \equiv (\widetilde{I}, \widetilde{r_I}, \widetilde{\pi}).$$

Remark 2.8 (Soundness Amplification)**.** Take any polynomial $\ell = \ell(\lambda, n)$. Then, given any NIZK in the hidden-bits model with soundness error ε, we can construct another NIZK in the hidden-bits model with ε^ℓ soundness error by running ℓ copies of the NIZK in parallel (and accepting a proof only if all of the ℓ copies are valid). The simulator would simulate each of the individual instances independently. Parallel repetition increases the length of the hidden-bits string by a factor of ℓ.

Theorem 2.9 (NIZKs in the Hidden-Bits Model [FLS99])**.** *For any $\varepsilon > 0$, every language $\mathcal{L} \in \mathsf{NP}$ has a NIZK in the hidden-bits model with soundness error ε and relying on a hidden-bits string of length $\rho = \mathsf{poly}(n, \log(1/\varepsilon))$.*

2.3 Designated-Verifier NIZKs and Dual-Mode NIZKs

We now review the notion of a *reusable* designated-verifier NIZK (DV-NIZK). Namely, we require that the same common reference string and verification state can be *reused* to prove and verify many statements without compromising either soundness or zero-knowledge. As in [LQR+19], we use the fine-grained notion with separate setup and key-generation algorithms. The setup algorithm samples the common reference string (CRS) while the key-generation algorithm generates a public key (used to generate proofs) along with a secret key (used to verify proofs). We allow the same CRS to be *reusable* by many verifiers, who each generate their own public/secret key-pairs. In the traditional notion of DV-NIZKs, the setup and key-generation algorithms would be combined into a single algorithm that outputs the CRS (which would include the public proving key) along with a secret verification key.

Definition 2.10 (Designated-Verifier NIZK)**.** Let $\mathcal{L} \subseteq \{0,1\}^n$ be an NP language associated with an NP relation $\mathcal{R}$ with $n = n(\lambda)$. A reusable *designated-verifier non-interactive zero-knowledge (DV-NIZK) proof for* $\mathcal{L}$ consists of a tuple of efficient algorithms $\Pi_{\mathsf{dvNIZK}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Prove}, \mathsf{Verify})$ with the following properties:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$: On input the security parameter λ, this algorithm outputs a common reference string crs. If Setup outputs a *uniformly random string*, we say that the scheme is in the *common random string* model.
- $\mathsf{KeyGen}(\mathsf{crs}) \to (\mathsf{pk}, \mathsf{sk})$: On input the common reference string crs, the key-generation algorithm outputs a public key pk and a secret key sk.
- $\mathsf{Prove}(\mathsf{crs}, \mathsf{pk}, x, w) \to \pi$: On input the common reference string crs, a public key pk, a statement $x \in \{0,1\}^n$, and a witness w, this algorithm outputs a proof π.
- $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x, \pi) \to \{0,1\}$: On input the common reference string crs, a secret verification key sk, a statement x, and a proof π, the verification algorithm outputs a bit $b \in \{0,1\}$.

Moreover, Π_{dvNIZK} should satisfy the following properties:

- **Completeness:** For all $(x, w) \in \mathcal{R}$, and taking $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$,
$$\Pr\left[\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, \mathsf{pk}, x, w) : \mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x, \pi) = 1\right] = 1.$$
- **(Statistical) soundness:** We consider two variants of soundness:
  - **Non-adaptive soundness:** For all $x \notin \mathcal{L}$ and all polynomials $q = q(\lambda)$, and all unbounded adversaries $\mathcal{A}$ making at most q verification queries, and sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$, we have that
$$\Pr\left[\pi \leftarrow \mathcal{A}^{\mathsf{Verify}(\mathsf{crs},\mathsf{sk},\cdot,\cdot)}(1^\lambda, \mathsf{crs}, \mathsf{pk}, x) : \mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x, \pi) = 1\right] = \mathsf{negl}(\lambda).$$
  - **Adaptive soundness:** For all polynomials $q = q(\lambda)$ and all unbounded adversaries $\mathcal{A}$ making at most q verification queries, and sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$, we have that
$$\Pr\left[(x, \pi) \leftarrow \mathcal{A}^{\mathsf{Verify}(\mathsf{crs},\mathsf{sk},\cdot,\cdot)}(1^\lambda, \mathsf{crs}, \mathsf{pk}) : x \notin \mathcal{L} \wedge \mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x, \pi) = 1\right] = \mathsf{negl}(\lambda).$$

  We also define the corresponding notions of *computational soundness* where the above properties only need to hold against efficient adversaries $\mathcal{A}$.
- **(Statistical) zero-knowledge:** For all polynomials $q = q(\lambda)$ and all unbounded adversaries $\mathcal{A}$ making at most q oracle queries, there exists an efficient simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that
$$\left|\Pr[\mathcal{A}^{\mathcal{O}_0(\mathsf{crs},\mathsf{pk},\cdot,\cdot)}(\mathsf{crs}, \mathsf{pk}, \mathsf{sk}) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_1(\mathsf{st}_\mathcal{S},\cdot,\cdot)}(\widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}}) = 1]\right| = \mathsf{negl}(\lambda),$$
  where $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$ and $(\mathsf{st}_\mathcal{S}, \widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}}) \leftarrow \mathcal{S}_1(1^\lambda)$, the oracle $\mathcal{O}_0(\mathsf{crs}, \mathsf{pk}, x, w)$ outputs $\mathsf{Prove}(\mathsf{crs}, \mathsf{pk}, x, w)$ if $\mathcal{R}(x, w) = 1$ and $\perp$ otherwise, and the oracle $\mathcal{O}_1(\mathsf{st}_\mathcal{S}, x, w)$ outputs $\mathcal{S}_2(\mathsf{st}_\mathcal{S}, x)$ if $\mathcal{R}(x, w) = 1$ and $\perp$ otherwise. Similar to soundness, we also consider *computational zero-knowledge* where the above property only needs to hold against efficient adversaries $\mathcal{A}$.

Definition 2.11 (Publicly-Verifiable NIZKs)**.** A NIZK Π_{NIZK} is *publicly-verifiable* if the secret key output by KeyGen is empty. In this case, we can combine the Setup and KeyGen algorithms into a single algorithm that just outputs the CRS, and there is no notion of separate public/secret keys pk and sk. Both the Prove and Verify algorithms just take crs as input. We can define all of the properties analogously. In the publicly-verifiable setting, we do not need to provide the prover a separate verification oracle in the soundness game.

Remark 2.12 (Single-Theorem vs. Multi-Theorem Zero-Knowledge)**.** The zero-knowledge property in Definition 2.10 is *multi-theorem* in the sense that the adversary can see proofs of multiple statements. We can consider a weaker notion of single-theorem zero-knowledge where the adversary can only see a proof on a single (adaptively-chosen) statement. Previously, Feige et al. [FLS99] showed how to generically compile a single-theorem NIZK into a multi-theorem NIZK using a PRG. This transformation also applies in the designated-verifier setting [QRW19, CH19, KNYY19a]. One limitation of the [FLS99] transformation is that it requires making *non-black-box* use of a PRG. The constructions we present in this work directly achieve multi-theorem zero-knowledge without needing to go through the [FLS99] transformation. As such, our constructions do *not* require making non-black-box use of any cryptographic primitives.

Malicious DV-NIZKs. We also consider the notion of a malicious designated-verifier NIZK (MDV-NIZK) from [QRW19] where zero-knowledge holds even when the public key pk is chosen maliciously. In this case, the only trusted setup that we require is generating the common reference string (or, in some cases, a common random string), which can be reused by many verifiers.

Definition 2.13 (Malicious Designated-Verifier NIZKs [QRW19])**.** Let $\Pi_{\mathsf{dvNIZK}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Prove}, \mathsf{Verify})$ be a DV-NIZK for a language $\mathcal{L}$. We say that Π_{dvNIZK} satisfies *statistical zero-knowledge against malicious verifiers* if for all polynomials $q = q(\lambda)$ and all unbounded adversaries $\mathcal{A}$ making at most q verification queries, there exists an efficient simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that

$$\left|\Pr[\mathsf{ExptReal}[\mathcal{A}, \mathcal{S}](1^\lambda)] - \Pr[\mathsf{ExptSim}[\mathcal{A}, \mathcal{S}](1^\lambda)]\right| = \mathsf{negl}(\lambda),$$

where the two experiments $\mathsf{ExptReal}[\mathcal{A}, \mathcal{S}](1^\lambda)$ and $\mathsf{ExptSim}[\mathcal{A}, \mathcal{S}](1^\lambda)$ proceed as follows:

- **Setup phase:** In $\mathsf{ExptReal}$, the challenger samples $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ and gives crs to $\mathcal{A}$. In $\mathsf{ExptSim}$, the challenger samples $(\mathsf{st}_{\mathcal{S}}, \widetilde{\mathsf{crs}}) \leftarrow \mathcal{S}_1(1^\lambda)$ and gives $\widetilde{\mathsf{crs}}$ to $\mathcal{A}$. The adversary replies with a public key pk.
- **Query phase:** Algorithm $\mathcal{A}$ is then given access to a verification oracle, and is allowed to make up to q queries to the oracle. On an input (x, w), the challenger replies with $\perp$ if $\mathcal{R}(x, w) \neq 1$. Otherwise, in $\mathsf{ExptReal}$, the challenger replies with $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, \mathsf{pk}, x, w)$ while in $\mathsf{ExptSim}$, the challenger replies with $\widetilde{\pi} \leftarrow \mathcal{S}_2(\mathsf{st}_{\mathcal{S}}, \mathsf{pk}, x)$.
- **Output phase:** At the end of the experiment, the adversary outputs a bit $b' \in \{0, 1\}$, which is the output of the experiment.

Correspondingly, we can define the analogous notion of computational zero-knowledge against malicious verifiers by only requiring the above property to hold against computationally-bounded adversaries. If Π_{dvNIZK} provides zero-knowledge against malicious verifiers, we say it is a *malicious-designated-verifier NIZK* (MDV-NIZK).

Dual-mode DV-NIZKs. Next, we recall the formal definition of a dual-mode (DV)-NIZK [GOS06, GOS12].

Definition 2.14 (Dual-Mode Designated-Verifier NIZK)**.** A dual-mode DV-NIZK $\Pi_{\mathsf{dvNIZK}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Prove}, \mathsf{Verify})$ is a DV-NIZK with the following additional properties:

- **Dual-mode:** The Setup algorithm takes an additional argument $\mathsf{mode} \in \{\mathsf{binding}, \mathsf{hiding}\}$, and outputs a common reference string crs.
- **CRS indistinguishability:** The common reference string output by the two modes are computationally indistinguishable:

$$\mathsf{Setup}(1^\lambda, \mathsf{binding}) \overset{c}{\approx} \mathsf{Setup}(1^\lambda, \mathsf{hiding}).$$

- **Statistical soundness in binding mode:** If $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{binding})$, the designated-verifier NIZK satisfies statistical soundness.
- **Statistical zero-knowledge in hiding mode:** If $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{hiding})$, the designated-verifier NIZK satisfies statistical zero-knowledge.

We define a dual mode MDV-NIZK analogously by requiring the stronger property of statistical zero-knowledge against malicious verifiers in hiding mode.

Remark 2.15 (Dual-Mode Designated-Verifier NIZKs)**.** Let $\Pi_{\mathsf{dvNIZK}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Prove}, \mathsf{Verify})$ be a dual-mode DV-NIZK for a language $\mathcal{L} \subseteq \{0,1\}^n$. Then, the following properties hold:

- When the CRS is generated in binding mode, Π_{dvNIZK} satisfies statistical soundness and computational zero-knowledge (i.e., Π_{dvNIZK} is a "computational DV-NIZK proof").
- When the CRS is generated in hiding mode, Π_{dvNIZK} satisfies *non-adaptive* computational soundness and statistical zero-knowledge (i.e., Π_{dvNIZK} is a "statistical DV-NIZK argument").
- If Π_{dvNIZK} is a dual-mode MDV-NIZK, then the zero-knowledge properties in each of the above instantiations also hold against malicious verifiers.

The first two properties follow from CRS indistinguishability and the corresponding statistical properties of Π_{dvNIZK} in the two modes. Note though that even if Π_{dvNIZK} satisfies adaptive soundness in binding mode, we do not know how to argue *adaptive soundness* for Π_{dvNIZK} in hiding mode. At a high-level, this is because in the definition of adaptive soundness, checking whether the adversary succeeded or not requires deciding whether the statement x output by the adversary is contained in the language $\mathcal{L}$ or not. Unless $\mathsf{NP} \subseteq \mathsf{P/poly}$, this is not an efficiently-checkable property in general, and as such, we are not able to directly argue adaptive soundness of the construction. We refer to [AF07] for more discussion on the challenges of using black-box reductions to argue adaptive soundness for statistical NIZK arguments.

Remark 2.16 (Adaptive Soundness via Complexity Leveraging)**.** Using complexity leveraging [BB04] and relying on a sub-exponential hardness assumption (as in [GOS06, GOS12]), we can show that non-adaptive soundness implies adaptive soundness. A direct application of complexity leveraging to a dual-mode NIZK yields an adaptively-sound statistical NIZK argument for proving statements of a priori bounded length $n = n(\lambda)$. Using the method from [QRW19, §7] (see also Remark 6.5), we can also obtain adaptive soundness for statements with arbitrary polynomial length, but still at the expense of a subexponential hardness assumption.

3 Dual-Mode Hidden-Bits Generators and Dual-Mode DV-NIZKs

In this section, we formally define a dual-mode hidden-bits generator. Our definition extends the notion of a hidden-bits generator from [QRW19] (and the similar notion of a designated-verifier PRG from [CH19]). Our definition differs from that in [QRW19] in the following respects:

- **Dual mode:** We require that the common reference string for the hidden-bits generator can be generated in two computationally indistinguishable modes: a *binding* mode where the commitment statistically binds to a sequence of hidden bits, and a *hiding* mode where the commitment (and the openings to any subset of the bits) statistically hide the remaining bits.
- **Statistical simulation in hiding mode.** Minimally, our hiding property requires that the commitment and openings to any subset of the bits output by the HBG *statistically* hide the unopened bits. Here, we require an even stronger *simulation* property where there is an efficient simulator that can simulate the commitment and openings to any (random) string, given *only* the values of the opened bits. Moreover, we allow the adversary to adaptively choose the subset of bits for which it wants to see openings, and we also allow *multiple* interactions with the simulator. This strong simulation property enables us to directly argue *adaptive* and *multi-theorem statistical zero-knowledge* for our NIZK constructions (Section 3.1).[7]

Definition 3.1 (Dual-Mode Hidden-Bits Generator)**.** Let λ be a security parameter and ρ be the output length. Let $\ell = \ell(\lambda, \rho)$ be a polynomial. A dual-mode (designated-verifier) hidden-bits generator (HBG) with commitments of length ℓ consists of a tuple of efficient algorithms $\Pi_{\mathsf{HBG}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{GenBits}, \mathsf{Verify})$ with the following properties:

- $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode}) \to \mathsf{crs}$: On input the security parameter λ, a length ρ, and a mode $\mathsf{mode} \in \{\mathsf{binding}, \mathsf{hiding}\}$, the setup algorithm outputs a common reference string crs.
- $\mathsf{KeyGen}(\mathsf{crs}) \to (\mathsf{pk}, \mathsf{sk})$: On input a common reference string crs, the key-generation algorithm outputs a public key pk and a secret key sk.
- $\mathsf{GenBits}(\mathsf{crs}, \mathsf{pk}) \to (\sigma, r, \{\pi_i\}_{i\in[\rho]})$: On input a common reference string crs and a public key pk, the bit-generation algorithm outputs a commitment $\sigma \in \{0,1\}^\ell$, a string $r \in \{0,1\}^\rho$, and a collection of proofs π_i for $i \in [\rho]$.
- $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i) \to \{0,1\}$: On input a common reference string crs, a secret key sk, a commitment $\sigma \in \{0,1\}^\ell$, an index $i \in [\rho]$, a bit $r_i \in \{0,1\}$, and a proof π_i, the verification algorithm outputs a bit $b \in \{0,1\}$.

In addition, we require that Π_{HBG} satisfy the following properties:

- **Correctness:** For all integers $\lambda \in \mathbb{N}$, and all polynomials $\rho = \rho(\lambda)$, all indices $i \in [\rho]$ and both modes $\mathsf{mode} \in \{\mathsf{binding}, \mathsf{hiding}\}$, and sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode})$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$, and $(\sigma, r, \{\pi_i\}_{i\in[\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs}, \mathsf{pk})$, we have

$$\Pr[\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i) = 1] = 1.$$

[7] The previous notion from [QRW19] was only sufficient for single-theorem non-adaptive computational zero-knowledge. Extending to adaptive multi-theorem computational zero-knowledge required imposing additional properties on the underlying NIZK in the hidden-bits model as well as making non-black-box use of cryptographic primitives [FLS99].

- **Succinctness:** The length ℓ of the commitment depends only on the security parameter and not the length of the output: namely, $\ell = \mathsf{poly}(\lambda)$.[8]

- **CRS indistinguishability:** For all polynomials $\rho = \rho(\lambda)$, we have that

$$\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{binding}) \overset{c}{\approx} \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding}).$$

- **Statistically binding in binding mode:** There exists a (possibly inefficient) deterministic algorithm $\mathsf{Open}(\mathsf{crs}, \sigma)$ such that for all polynomials $\rho = \rho(\lambda)$ and $q = q(\lambda)$ and all unbounded adversaries $\mathcal{A}$ making up to q oracle queries, and sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{binding})$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$, $(\sigma^*, i^*, r^*, \pi^*) \leftarrow \mathcal{A}^{\mathsf{Verify}(\mathsf{crs},\mathsf{sk},\cdot,\cdot,\cdot,\cdot)}(1^\lambda, 1^\rho, \mathsf{crs}, \mathsf{pk})$, $r \leftarrow \mathsf{Open}(\mathsf{crs}, \sigma^*)$, we have that

$$\Pr[r_{i^*} \neq r^* \wedge \mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma^*, i^*, r^*, \pi^*) = 1] = \mathsf{negl}(\lambda).$$

- **Statistical simulation in hiding mode:** For all polynomials $\rho = \rho(\lambda)$, $q = q(\lambda)$, and all unbounded adversaries $\mathcal{A}$ making up to q queries, there exists an efficient simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that

$$\left| \Pr[\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 0](1^\lambda, 1^\rho) = 1] - \Pr[\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 1](1^\lambda, 1^\rho) = 1] \right| = \mathsf{negl}(\lambda), \quad (3.1)$$

  where for a bit $b \in \{0,1\}$, the hiding experiment $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, b](1^\lambda, 1^\rho)$ is defined as follows:

  - **Setup phase:** If $b = 0$, the challenger samples $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$, and gives $(\mathsf{crs}, \mathsf{pk}, \mathsf{sk})$ to $\mathcal{A}$. If $b = 1$, it samples $(\mathsf{st}_\mathcal{S}, \widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}}) \leftarrow \mathcal{S}_1(1^\lambda, 1^\rho)$ and gives $(\widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}})$ to $\mathcal{A}$.
  - **Query phase:** The adversary $\mathcal{A}$ can now make up to q challenge queries. On each query, the challenger responds as follows:
    - If $b = 0$, the challenger computes $(\sigma, r, \{\pi_i\}_{i\in[\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs}, \mathsf{pk})$ and gives r to the adversary. If $b = 1$, the challenger responds with $\widetilde{r} \xleftarrow{R} \{0,1\}^\rho$.
    - The adversary specifies a subset $I \subseteq [\rho]$.
    - If $b = 0$, then the challenger replies with the pair $(\sigma, \{\pi_i\}_{i\in[I]})$ it sampled above. If $b = 1$, the challenger replies to $\mathcal{A}$ with $(\widetilde{\sigma}, \{\widetilde{\pi}_i\}_{i\in I}) \leftarrow \mathcal{S}_2(\mathsf{st}_\mathcal{S}, I, \widetilde{r}_I)$.
  - **Output phase:** At the end of the experiment, the adversary outputs a bit $b \in \{0,1\}$, which is the output of the experiment.

  When the difference in Eq. (3.1) is identically zero, we say that Π_{HBG} satisfies *perfect simulation in hiding mode*.

Definition 3.2 (Publicly-Verifiable Dual-Mode HBG)**.** A dual-mode HBG Π_{HBG} is publicly-verifiable if the secret key sk output by KeyGen is empty. In this case, we can combine the Setup algorithm and the KeyGen algorithm into a single algorithm that just outputs the crs, and there is no notion of separate public/secret keys pk and sk. The $\mathsf{GenBits}$ and Verify algorithms just take crs as input. We define all of the other properties analogously. In the publicly-verifiable setting, we do not need to provide the verification oracle to the adversary in the statistical binding security definition.

[8] We remark that this is a *stronger* requirement than the corresponding requirement in [QRW19], which also allows ℓ to scale sublinearly with ρ. We use this definition because it is conceptually simpler and all of our constructions satisfy this stronger property.

Definition 3.3 (Statistical Simulation for Malicious Keys)**.** Let $\Pi_{\mathsf{HBG}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{GenBits}, \mathsf{Verify})$ be a hidden-bits generator. We say that Π_{HBG} satisfies *statistical simulation for malicious keys* if Π_{HBG} satisfies the following stronger simulation property (where the adversary chooses the public key pk) in hiding mode:

- **Statistical simulation for malicious keys:** For all polynomials $\rho = \rho(\lambda)$, $q = q(\lambda)$, and all unbounded adversaries $\mathcal{A}$ making up to q queries, there exists an efficient simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that

$$\left| \Pr[\mathsf{ExptHide}^*[\mathcal{A}, \mathcal{S}, 0](1^\lambda, 1^\rho) = 1] - \Pr[\mathsf{ExptHide}^*[\mathcal{A}, \mathcal{S}, 1](1^\lambda, 1^\rho) = 1] \right| = \mathsf{negl}(\lambda),$$

  where for a bit $b \in \{0,1\}$, the hiding experiment $\mathsf{ExptHide}^*[\mathcal{A}, \mathcal{S}, b](1^\lambda, 1^\rho)$ is defined to be $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, b](1^\lambda, 1^\rho)$ with the following differences:

  - **Setup phase:** If $b = 0$, the challenger samples $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$ and gives crs to $\mathcal{A}$. If $b = 1$, the challenger samples $(\mathsf{st}_{\mathcal{S}}, \widetilde{\mathsf{crs}}) \leftarrow \mathcal{S}_1(1^\lambda, 1^\rho)$ and gives $\widetilde{\mathsf{crs}}$ to $\mathcal{A}$. The adversary then chooses a public key pk.
  - **Query phase:** Same as in $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, b]$, except when $b = 1$, the challenger also provides the (adversarially-chosen) public key pk to the simulator. In other words, when $b = 1$, the challenger's reply to $\mathcal{A}$ is computed as $(\widetilde{\sigma}, \{\widetilde{\pi}_i\}_{i \in I}) \leftarrow \mathcal{S}_2(\mathsf{st}_{\mathcal{S}}, \mathsf{pk}, I, \widetilde{r}_I)$.
  - **Output phase:** Same as in $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, b]$.

3.1 Dual-Mode DV-NIZK from Dual-Mode HBG

In this section, we give our construction of a dual-mode designated-verifier NIZK from a dual-mode designated-verifier HBG and a NIZK in the hidden-bits model. Our generic construction is essentially the same as the corresponding construction from [QRW19]. We do rely on a different argument to show adaptive, multi-theorem statistical zero-knowledge, and in particular, we appeal to the statistical simulation property of our dual-mode HBG that we introduced in Definition 3.1.

Construction 3.4 (Dual-Mode DV-NIZK from Dual-Mode HBG)**.** Let $\mathcal{L} \subseteq \{0,1\}^n$ be an NP language with associated NP relation $\mathcal{R}$. We rely on the following building blocks:

- Let $\Pi_{\mathsf{HBM}} = (\mathsf{HBM.Prove}, \mathsf{HBM.Verify})$ be a NIZK in the hidden-bits model for $\mathcal{L}$, and let $\rho = \rho(\lambda)$ be the length of the hidden-bits string for Π_{HBM}.
- Let $\Pi_{\mathsf{HBG}} = (\mathsf{HBG.Setup}, \mathsf{HBG.KeyGen}, \mathsf{HBG.GenBits}, \mathsf{HBG.Verify})$ be a hidden-bits generator with commitments of length $\ell = \ell(\lambda, \rho)$, where λ is the security parameter and ρ is the output length of the generator.

We construct a dual-mode DV-NIZK $\Pi_{\mathsf{dvNIZK}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Prove}, \mathsf{Verify})$ for $\mathcal{L}$ as follows:

- $\mathsf{Setup}(1^\lambda, \mathsf{mode}) \to \mathsf{crs}$: On input λ and $\mathsf{mode} \in \{\mathsf{binding}, \mathsf{hiding}\}$, sample $s \xleftarrow{R} \{0,1\}^\rho$. Then, run $\mathsf{crs}_{\mathsf{HBG}} \leftarrow \mathsf{HBG.Setup}(1^\lambda, 1^\rho, \mathsf{mode})$, and output $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$.
- $\mathsf{KeyGen}(\mathsf{crs}) \to (\mathsf{pk}, \mathsf{sk})$: On input $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$, the key-generation algorithm runs $(\mathsf{pk}_{\mathsf{HBG}}, \mathsf{sk}_{\mathsf{HBG}}) \leftarrow \mathsf{HBG.KeyGen}(\mathsf{crs}_{\mathsf{HBG}})$ and outputs $\mathsf{pk} = \mathsf{pk}_{\mathsf{HBG}}$ and $\mathsf{sk} = \mathsf{sk}_{\mathsf{HBG}}$.

- $\mathsf{Prove}(\mathsf{crs}, \mathsf{pk}, x, w) \to \pi$: On input $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$, $\mathsf{pk} = \mathsf{pk}_{\mathsf{HBG}}$, $x \in \{0,1\}^n$, and w, compute a hidden-bits string $(\sigma, r, \{\pi_{\mathsf{HBG},i}\}_{i\in[\rho]}) \leftarrow \mathsf{HBG.GenBits}(\mathsf{crs}_{\mathsf{HBG}}, \mathsf{pk}_{\mathsf{HBG}})$, and an HBM proof $(I, \pi_{\mathsf{HBM}}) \leftarrow \mathsf{HBM.Prove}(1^\lambda, r \oplus s, x, w)$. Output $\pi = (\sigma, I, r_I, \{\pi_{\mathsf{HBG},i}\}_{i\in I}, \pi_{\mathsf{HBM}})$.
- $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x, \pi)$: On input $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$, $\mathsf{sk} = \mathsf{sk}_{\mathsf{HBG}}$, $x \in \{0,1\}^n$, and the proof $\pi = (\sigma, I, r_I, \{\pi_{\mathsf{HBG},i}\}_{i\in I}, \pi_{\mathsf{HBM}})$, output 1 if $\mathsf{HBM.Verify}(1^\lambda, I, r_I \oplus s_I, x, \pi_{\mathsf{HBM}}) = 1$ and $\mathsf{HBG.Verify}(\mathsf{crs}_{\mathsf{HBG}}, \mathsf{sk}_{\mathsf{HBG}}, \sigma, i, r_i, \pi_{\mathsf{HBG},i}) = 1$ for all $i \in I$. Otherwise, output 0.

Theorem 3.5 (Completeness)**.** *If Π_{HBM} is complete and Π_{HBG} is correct, then Π_{dvNIZK} from Construction 3.4 is complete.*

Proof. Take any $\mathsf{mode} \in \{\mathsf{binding}, \mathsf{hiding}\}$, and sample $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{mode})$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$. Here, $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$, $\mathsf{pk} = \mathsf{pk}_{\mathsf{HBG}}$, and $\mathsf{sk} = \mathsf{sk}_{\mathsf{HBG}}$. Take any statement $(x, w) \in \mathcal{R}$, and let $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, \mathsf{pk}, x, w)$. Then $\pi = (\sigma, I, r_I, \{\pi_{\mathsf{HBG},i}\}_{i\in I}, \pi_{\mathsf{HBM}})$. Consider the behavior of $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x, \pi)$. By correctness of Π_{HBG}, $\mathsf{HBG.Verify}(\mathsf{crs}_{\mathsf{HBG}}, \mathsf{sk}_{\mathsf{HBG}}, \sigma, i, r_i, \pi_{\mathsf{HBG},i}) = 1$ for all $i \in I$. By completeness of Π_{HBM}, $\mathsf{HBM.Verify}(1^\lambda, I, r_I \oplus s_I, x, w) = 1$, and the verifier accepts. □

Theorem 3.6 (CRS Indistinguishability)**.** *If Π_{HBG} satisfies CRS indistinguishability, then Π_{dvNIZK} from Construction 3.4 satisfies CRS indistinguishability.*

Proof. The CRS in Construction 3.4 consists of a tuple $(\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$. In both modes, the first two components are identically distributed, and $\mathsf{crs}_{\mathsf{HBG}}$ is computationally indistinguishable by CRS indistinguishability of Π_{HBG}. □

Theorem 3.7 (Statistical Soundness in Binding Mode)**.** *If Π_{HBM} is statistically sound with soundness error $\varepsilon(\lambda)$, Π_{HBG} is statistically binding in binding mode, and $2^\ell \cdot \varepsilon = \mathsf{negl}(\lambda)$ then Π_{dvNIZK} from Construction 3.4 satisfies adaptive statistical soundness.*

The proof of Theorem 3.7 is very similar to the corresponding proof of adaptive statistical soundness from [QRW19]. We include it in Appendix A.

Theorem 3.8 (Statistical Zero-Knowledge in Hiding Mode)**.** *If Π_{HBM} satisfies statistical (resp., perfect) zero-knowledge and Π_{HBG} provides statistical (resp., perfect) simulation in hiding mode, then Π_{dvNIZK} from Construction 3.4 satisfies statistical (resp., perfect) zero-knowledge in hiding mode.*

Proof. Let $\mathcal{S}_{\mathsf{HBM}}$ denote the zero-knowledge simulator for Π_{HBM} and $\mathcal{S}_{\mathsf{HBG}} = (\mathcal{S}_{\mathsf{HBG},1}, \mathcal{S}_{\mathsf{HBG},2})$ be the simulator for Π_{HBG} in hiding mode. We construct a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ as follows:

- $\mathcal{S}_1(1^\lambda) \to (\mathsf{st}_{\mathcal{S}}, \widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}})$: Run $(\mathsf{st}_{\mathsf{HBG}}, \widetilde{\mathsf{crs}}_{\mathsf{HBG}}, \widetilde{\mathsf{pk}}_{\mathsf{HBG}}, \widetilde{\mathsf{sk}}_{\mathsf{HBG}}) \leftarrow \mathcal{S}_{\mathsf{HBG},1}(1^\lambda, 1^\rho)$. Choose $\widetilde{s} \xleftarrow{R} \{0,1\}^\rho$ and set $\mathsf{st}_{\mathcal{S}} = \mathsf{st}_{\mathsf{HBG}}$, $\widetilde{\mathsf{crs}} = (\lambda, s, \widetilde{\mathsf{crs}}_{\mathsf{HBG}})$, $\widetilde{\mathsf{pk}} = \widetilde{\mathsf{pk}}_{\mathsf{HBG}}$, and $\widetilde{\mathsf{sk}} = \widetilde{\mathsf{sk}}_{\mathsf{HBG}}$. Output $(\mathsf{st}_{\mathcal{S}}, \widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}})$.
- $\mathcal{S}_2(\mathsf{st}_{\mathcal{S}}, x) \to \widetilde{\pi}$: On input $\mathsf{st}_{\mathcal{S}} = \mathsf{st}_{\mathsf{HBG}}$ and $x \in \{0,1\}^n$, run $(\widetilde{I}, \widetilde{r}_{\widetilde{I}}, \widetilde{\pi}_{\mathsf{HBM}}) \leftarrow \mathcal{S}_{\mathsf{HBM}}(1^\lambda, x)$ and $(\widetilde{\sigma}, \{\widetilde{\pi}_{\mathsf{HBG},i}\}_{i\in\widetilde{I}}) \leftarrow \mathcal{S}_{\mathsf{HBG},2}(\mathsf{st}_{\mathsf{HBG}}, \widetilde{I}, \widetilde{r}_{\widetilde{I}} \oplus \widetilde{s}_{\widetilde{I}})$. Output the simulated proof $\widetilde{\pi} = (\widetilde{\sigma}, \widetilde{I}, \widetilde{r}_{\widetilde{I}} \oplus \widetilde{s}_{\widetilde{I}}, \{\widetilde{\pi}_{\mathsf{HBG},i}\}_{i\in\widetilde{I}}, \widetilde{\pi}_{\mathsf{HBM}})$.

To complete the proof, we proceed via a hybrid argument:

- Hyb_0: This is the real distribution. Namely, the challenger begins by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$, and gives $(\mathsf{crs}, \mathsf{pk}, \mathsf{sk})$ to the adversary. The challenger responds to oracle queries on inputs (x, w) with $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, \mathsf{pk}, x, w)$ if $\mathcal{R}(x, w) = 1$ and with $\perp$ otherwise.

  In more detail, the challenger samples $s \xleftarrow{R} \{0,1\}^\rho$ and runs $\mathsf{crs}_{\mathsf{HBG}} \leftarrow \mathsf{HBG.Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$, and $(\mathsf{pk}_{\mathsf{HBG}}, \mathsf{sk}_{\mathsf{HBG}}) \leftarrow \mathsf{HBG.KeyGen}(\mathsf{crs}_{\mathsf{HBG}})$. It sets $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$, $\mathsf{pk} = \mathsf{pk}_{\mathsf{HBG}}$, and $\mathsf{sk} = \mathsf{sk}_{\mathsf{HBG}}$. When the adversary makes an oracle query on (x, w) where $\mathcal{R}(x, w) = 1$, the challenger computes $(\sigma, r, \{\pi_{\mathsf{HBG},i}\}_{i\in[\rho]}) \leftarrow \mathsf{HBG.GenBits}(\mathsf{crs}_{\mathsf{HBG}}, \mathsf{pk}_{\mathsf{HBG}})$ and $(I, \pi_{\mathsf{HBM}}) \leftarrow \mathsf{HBM.Prove}(1^\lambda, r \oplus s, x, w)$. It outputs $\pi = (\sigma, I, r_I, \{\pi_{\mathsf{HBG},i}\}_{i\in I}, \pi_{\mathsf{HBM}})$.

- Hyb_1: Same as Hyb_0, except the challenger uses $\mathcal{S}_{\mathsf{HBG},1}$ to generate the common reference string and public/secret keys. It uses $\mathcal{S}_{\mathsf{HBG},2}$ to simulate the openings to the hidden-bits generator when responding to oracle queries. Specifically, the challenger works as follows:

  1. At the start of the game, the challenger runs $(\mathsf{st}_{\mathsf{HBG}}, \widetilde{\mathsf{crs}}_{\mathsf{HBG}}, \widetilde{\mathsf{pk}}_{\mathsf{HBG}}, \widetilde{\mathsf{sk}}_{\mathsf{HBG}}) \leftarrow \mathcal{S}_{\mathsf{HBG},1}(1^\lambda, 1^\rho)$. It samples $\widetilde{s} \xleftarrow{R} \{0,1\}^\rho$ and gives $\widetilde{\mathsf{crs}} = (\lambda, \widetilde{s}, \widetilde{\mathsf{crs}}_{\mathsf{HBG}})$, $\widetilde{\mathsf{pk}} = \widetilde{\mathsf{pk}}_{\mathsf{HBG}}$, and $\widetilde{\mathsf{sk}} = \widetilde{\mathsf{sk}}_{\mathsf{HBG}}$ to the adversary.
  2. Whenever the adversary makes an oracle query (x, w) where $\mathcal{R}(x, w) = 0$, the challenger replies with $\perp$. Otherwise, it samples $r \xleftarrow{R} \{0,1\}^\rho$ and runs $(I, \pi_{\mathsf{HBM}}) \leftarrow \mathsf{HBM.Prove}(1^\lambda, r \oplus \widetilde{s}, x, w)$. Then, it samples $(\widetilde{\sigma}, \{\widetilde{\pi}_{\mathsf{HBG},i}\}_{i\in I}) \leftarrow \mathcal{S}_{\mathsf{HBG},2}(\mathsf{st}_{\mathsf{HBG}}, I, r_I)$, and finally outputs the proof $\pi = (\widetilde{\sigma}, I, r_I, \{\widetilde{\pi}_{\mathsf{HBG},i}\}_{i\in I}, \pi_{\mathsf{HBM}})$.

- Hyb_2: Same as Hyb_1, except when responding to oracle queries, the challenger uses the simulator for the hidden-bits model NIZK to simulate the proofs. More precisely, on input (x, w) where $\mathcal{R}(x, w) = 1$, the challenger computes $(\widetilde{I}, \widetilde{r}'_{\widetilde{I}}, \widetilde{\pi}_{\mathsf{HBM}}) \leftarrow \mathcal{S}_{\mathsf{HBM}}(1^\lambda, x)$. It sets $\widetilde{r}_{\widetilde{I}} = \widetilde{r}'_{\widetilde{I}} \oplus \widetilde{s}_{\widetilde{I}}$, and computes $(\widetilde{\sigma}, \{\widetilde{\pi}_{\mathsf{HBG},i}\}_{i\in I}) \leftarrow \mathcal{S}_{\mathsf{HBG},2}(\mathsf{st}_{\mathsf{HBG}}, \widetilde{I}, \widetilde{r}_{\widetilde{I}})$ and outputs the proof $\widetilde{\pi} = (\widetilde{\sigma}, \widetilde{I}, \widetilde{r}_{\widetilde{I}}, \{\widetilde{\pi}_{\mathsf{HBG},i}\}_{i\in\widetilde{I}}, \widetilde{\pi}_{\mathsf{HBM}})$. This is the simulated distribution (up to interchanging the labels $\widetilde{r}_{\widetilde{I}}$ and $\widetilde{r}'_{\widetilde{I}}$).

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of Hyb_i with adversary $\mathcal{A}$. We now show that the output distributions of each adjacent pair of hybrid experiments are statistically indistinguishable.

Lemma 3.9. *If Π_{HBG} satisfies statistical (resp., perfect) simulation in hiding mode, then the output distributions of Hyb_0 and Hyb_1 are statistically (resp., perfectly) indistinguishable.*

Proof. Suppose there exists an adversary $\mathcal{A}$ such that $|\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1]| = \varepsilon$ for some non-negligible ε. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ such that $\mathsf{ExptHide}[\mathcal{B}, \mathcal{S}_{\mathsf{HBG}}, 0]$ and $\mathsf{ExptHide}[\mathcal{B}, \mathcal{S}_{\mathsf{HBG}}, 1]$ are distinguishable (with the same advantage ε). Algorithm $\mathcal{B}$ works as follows:

- First, algorithm $\mathcal{B}$ receives a tuple $(\mathsf{crs}_{\mathsf{HBG}}, \mathsf{pk}_{\mathsf{HBG}}, \mathsf{sk}_{\mathsf{HBG}})$ from the challenger. It samples $s \xleftarrow{R} \{0,1\}^\rho$, and gives $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$, $\mathsf{pk} = \mathsf{pk}_{\mathsf{HBG}}$, and $\mathsf{sk} = \mathsf{sk}_{\mathsf{HBG}}$ to $\mathcal{A}$.

- When $\mathcal{A}$ makes a query on (x, w), if $\mathcal{R}(x, w) = 0$, algorithm $\mathcal{B}$ responds with $\perp$. Otherwise, $\mathcal{B}$ makes a challenge query to its challenger to obtain a string $r \in \{0,1\}^\rho$. Algorithm $\mathcal{B}$ computes $(I, \pi_{\mathsf{HBM}}) \leftarrow \mathsf{HBM.Prove}(1^\lambda, r \oplus s, x, w)$ and gives $I \subseteq [\rho]$ to the challenger.

The challenger replies with a pair $(\sigma, \{\pi_{\mathsf{HBG},i}\}_{i\in I})$. Finally, $\mathcal{B}$ replies to $\mathcal{A}$ with the proof $\pi = (\sigma, I, r_I, \{\pi_{\mathsf{HBG},i}\}_{i\in I}, \pi_{\mathsf{HBM}})$.

- At the end of the game, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs.

By construction, if the challenger generates the parameters and answers the queries according to $\mathsf{ExptHide}[\mathcal{B}, \mathcal{S}_{\mathsf{HBG}}, 0]$, then $\mathcal{B}$ perfectly simulates Hyb_0 for $\mathcal{A}$. Alternatively, if the challenger implements the logic according to $\mathsf{ExptHide}[\mathcal{B}, \mathcal{S}_{\mathsf{HBG}}, 1]$, then $\mathcal{B}$ perfectly simulates Hyb_1 for $\mathcal{A}$. □

Lemma 3.10. *If Π_{HBM} satisfies statistical (resp., perfect) zero-knowledge, then the output distributions of Hyb_1 and Hyb_2 are statistically (resp., perfectly) indistinguishable.*

Proof. We define a sequence of $q+1$ intermediate hybrid experiments $\mathsf{Hyb}_{1,0}, \ldots, \mathsf{Hyb}_{1,q}$, where $q = \mathsf{poly}(\lambda)$ is a bound on the number of queries the adversary makes. Hybrid $\mathsf{Hyb}_{1,i}$ is the experiment where the first i oracle queries are handled according to the procedure in Hyb_2 and the remaining queries are handled according to the procedure in Hyb_1. By construction $\mathsf{Hyb}_{1,0} \equiv \mathsf{Hyb}_1$ and $\mathsf{Hyb}_{1,q} \equiv \mathsf{Hyb}_2$. Moreover, each adjacent pair of hybrid experiments $\mathsf{Hyb}_{1,i-1}$ and $\mathsf{Hyb}_{1,i}$ only differ in how the i^{th} oracle query (x, w) is handled. In $\mathsf{Hyb}_{1,i-1}$, the challenger samples a random $r \xleftarrow{R} \{0,1\}^\rho$ and computes $(I, \pi_{\mathsf{HBM}}) \leftarrow \mathsf{HBM.Prove}(1^\lambda, r \oplus \widetilde{s}, x, w)$ while in $\mathsf{Hyb}_{1,i}$, the challenger invokes the simulator $(\widetilde{I}, \widetilde{r}'_{\widetilde{I}}, \widetilde{\pi}_{\mathsf{HBM}}) \leftarrow \mathcal{S}_{\mathsf{HBM}}(1^\lambda, x)$. By statistical zero-knowledge of Π_{HBM}, we have that $(I, r_I \oplus \widetilde{s}_I, \pi_{\mathsf{HBM}}) \overset{s}{\approx} (\widetilde{I}, \widetilde{r}'_{\widetilde{I}}, \widetilde{\pi}_{\mathsf{HBM}})$. If Π_{HBM} satisfies perfect zero-knowledge, then these two distributions are identically distributed. Correspondingly, hybrids $\mathsf{Hyb}_{1,i-1}$ and $\mathsf{Hyb}_{1,i}$ are statistically indistinguishable (or identically distributed if Π_{HBM} satisfies perfect zero-knowledge). Since $q = \mathsf{poly}(\lambda)$, we conclude by a hybrid argument that the outputs of Hyb_2 and Hyb_3 are statistically indistinguishable (or identically distributed if Π_{HBM} satisfies perfect zero-knowledge). □

Since each consecutive pair of hybrid experiments is statistically indistinguishable (or identically distributed), the claim follows. □

Theorem 3.11 (Statistical Zero-Knowledge against Malicious Verifiers)**.** *If Π_{HBM} satisfies statistical zero-knowledge and Π_{HBG} provides statistical simulation for malicious keys, then Construction 3.4 is a MDV-NIZK. Namely, Construction 3.4 satisfies statistical zero-knowledge against malicious verifiers in hiding mode.*

The proof of Theorem 3.11 follows from a similar argument as Theorem 3.8 and is included in Appendix A.

4 Dual-Mode HBGs from the k-Lin Assumption

In this section, we show how to construct dual-mode hidden-bits generators from the k-Lin assumption. We begin with a basic construction from the k-Lin assumption (Section 4.1) and then show how to extend it to achieve public verifiability in a pairing group (Section 4.2) as well as how to achieve security against malicious verifiers in a pairing-free group (Section 4.3).

4.1 Dual-Mode Hidden-Bits Generator from k-Lin

In this section, we show how to construct a dual-mode hidden-bits generator from the k-linear (k-Lin) assumption [BBS04, HK07, Sha07, EHK+13] over *pairing-free* groups for any $k \geq 1$. We note that the 1-Lin assumption is precisely the decisional Diffie-Hellman (DDH) assumption. We begin by recalling some basic notation as well as the k-Lin assumption.

Notation. Throughout this section, we will work with cyclic groups $\mathbb{G}$ of prime order p. We will use multiplicative notation to denote the group operation. For $x \in \mathbb{Z}_p$, we often refer to g^x as an "encoding" of x. For a matrix $\mathbf{A} \in \mathbb{Z}_p^{n \times m}$, we write $g^{\mathbf{A}} \in \mathbb{G}^{n \times m}$ to denote the matrix of group elements formed by taking the element-wise encoding of each component of $\mathbf{A}$.

Definition 4.1 (Prime-Order Group Generator)**.** A prime-order group generator algorithm $\mathsf{GroupGen}$ is an efficient algorithm that on input the security parameter 1^λ outputs a description $\mathcal{G} = (\mathbb{G}, p, g)$ of a prime-order group $\mathbb{G}$ with order p and generator g. Throughout this work, we will assume that $1/p = \mathsf{negl}(\lambda)$.

Definition 4.2 (k-Linear Assumption [BBS04, HK07, Sha07, EHK+13])**.** Fix a constant $k \geq 1$. We say that a prime-order group generator $\mathsf{GroupGen}$ satisfies the k-Lin assumption if for all efficient adversaries $\mathcal{A}$ and sampling $\mathcal{G} = (\mathbb{G}, p, g) \leftarrow \mathsf{GroupGen}(1^\lambda)$, $\mathbf{a} \xleftarrow{R} \mathbb{Z}_p^k$, $\mathbf{w} \xleftarrow{R} \mathbb{Z}_p^{k+1}$, and $\mathbf{u} \xleftarrow{R} \mathbb{Z}_p^k$, we have that

$$\left|\Pr[\mathcal{A}(\mathcal{G}, g^{\mathbf{A}}, g^{\mathbf{Aw}}) = 1] - \Pr[\mathcal{A}(\mathcal{G}, g^{\mathbf{A}}, g^{\mathbf{u}}) = 1]\right| = \mathsf{negl}(\lambda),$$

where

$$\mathbf{A} = \begin{pmatrix} \mathrm{diag}(\mathbf{a}) \\ \hline \mathbf{1}^\mathsf{T} \end{pmatrix} \in \mathbb{Z}_p^{(k+1) \times k}, \tag{4.1}$$

and $\mathrm{diag}(\mathbf{a}) \in \mathbb{Z}_p^{k \times k}$ denotes the diagonal matrix whose entries are $a_1, \ldots, a_k$, and $\mathbf{1} \in \mathbb{Z}_p^{k+1}$ is the all-ones vector.

Construction 4.3 (Dual-Mode Hidden-Bits Generator from k-Lin)**.** Let $\mathsf{GroupGen}$ be a prime-order group generator algorithm. We construct a dual-mode hidden-bits generator (HBG) as follows:

- $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode}) \to \mathsf{crs}$: First, the setup algorithm samples $\mathcal{G} = (\mathbb{G}, p, g) \leftarrow \mathsf{GroupGen}(1^\lambda)$ and a hash function $H \xleftarrow{R} \mathcal{H}$, where $\mathcal{H}$ is a family of hash functions with domain $\mathbb{G}$ and range $\{0,1\}$. Next, it samples $\mathbf{V} \xleftarrow{R} \mathbb{Z}_p^{(\rho+k) \times k}$ and vectors $\mathbf{w}_1, \ldots, \mathbf{w}_\rho \in \mathbb{Z}_p^{\rho+k}$ as follows:
  - If $\mathsf{mode} = \mathsf{hiding}$, sample $\mathbf{w}_i \xleftarrow{R} \mathbb{Z}_p^{\rho+k}$ for all $i \in [\rho]$.
  - If $\mathsf{mode} = \mathsf{binding}$, sample $\mathbf{s}_i \xleftarrow{R} \mathbb{Z}_p^k$ and set $\mathbf{w}_i \leftarrow \mathbf{V}\mathbf{s}_i$ for all $i \in [\rho]$.

  Output $\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho})$.
- $\mathsf{KeyGen}(\mathsf{crs}) \to (\mathsf{pk}, \mathsf{sk})$: On input $\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho})$, the key-generation algorithm samples $a \xleftarrow{R} \mathbb{Z}_p$ and $\mathbf{b}_1, \ldots, \mathbf{b}_\rho \xleftarrow{R} \mathbb{Z}_p^k$. For each $i \in [\rho]$, it sets $\mathbf{z}_i \leftarrow \mathbf{w}_i a + \mathbf{V}\mathbf{b}_i \in \mathbb{Z}_p^{\rho+k}$. It outputs
$$\mathsf{pk} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_\rho}) \quad \text{and} \quad \mathsf{sk} = (a, \mathbf{b}_1, \ldots, \mathbf{b}_\rho).$$

- $\mathsf{GenBits}(\mathsf{crs}, \mathsf{pk}) \to (\sigma, r, \{\pi_i\}_{i\in[\rho]})$: On input $\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho})$ and public key $\mathsf{pk} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_\rho})$, sample $\mathbf{y} \xleftarrow{R} \mathbb{Z}_p^{\rho+k}$ and compute for each $i \in [\rho]$,

$$g^{t_i} \leftarrow g^{\mathbf{y}^\top \mathbf{w}_i} \quad \text{and} \quad g^{u_i} \leftarrow g^{\mathbf{y}^\top \mathbf{z}_i}.$$

  Next, let $\sigma = g^{\mathbf{y}^\top \mathbf{V}}$. For each $i \in [\rho]$, set $r_i \leftarrow H(g^{t_i})$ and $\pi_i \leftarrow (g^{t_i}, g^{u_i})$, and output σ, r, and $\{\pi_i\}_{i\in[\rho]}$.

- $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$: On input $\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho})$, secret key $\mathsf{sk} = (a, \mathbf{b}_1, \ldots, \mathbf{b}_\rho)$, $\sigma = g^{\mathbf{c}^\top}$, $i \in [\rho]$, $r_i \in \{0,1\}$, and $\pi_i = (g^{t_i}, g^{u_i})$, output 1 if $g^{u_i} = (g^{t_i a})(g^{\mathbf{c}^\top \mathbf{b}_i})$ and $r_i = H(g^{t_i})$. Otherwise, output 0.

Correctness and security analysis. We now state the correctness and security theorems for Construction 4.3 and give the proofs in Appendix C.1.

Theorem 4.4 (Correctness)**.** *Construction 4.3 is correct.*

Theorem 4.5 (Succinctness)**.** *Construction 4.3 is succinct.*

Theorem 4.6 (CRS Indistinguishability)**.** *Suppose the k-*Lin *assumption holds for* GroupGen*. Then, Construction 3.4 satisfies CRS indistinguishability.*

Theorem 4.7 (Statistical Binding in Binding Mode)**.** *Construction 4.3 satisfies statistical binding in binding mode.*

Theorem 4.8 (Statistical Simulation in Hiding Mode)**.** *If $\mathcal{H}$ satisfies statistical uniformity, then Construction 4.3 satisfies statistical simulation in hiding mode.*

Remark 4.9 (Common Random String in Hiding Mode)**.** Construction 4.3 has the property that in hiding mode, the CRS is a collection of *uniformly* random group elements; in other words, the CRS in hiding mode can be sampled as a common *random* string. In conjunction with Construction 3.4, we obtain a statistical NIZK argument in the common *random* string model (and a computational NIZK proof in the common *reference* string model).

4.2 Publicly-Verifiable Hidden-Bit Generators from Pairings

In this section, we describe a variant of our dual-mode hidden-bits generator from Section 4.1 to obtain a *publicly-verifiable* hidden-bits generator from pairings. Our resulting construction does not give a dual-mode hidden-bits generator. Instead, we obtain a standard HBG (where there is a *single* mode) that satisfies statistical simulation and computational binding. Using an analog of Construction 3.4, this suffices to construct a publicly-verifiable statistical NIZK argument. We refer to Appendix B for the details. Below, we define the computational binding property we use:

Definition 4.10 (Computational Binding)**.** A *publicly-verifiable* hidden bits generator $\Pi_{\mathsf{HBG}} = (\mathsf{Setup}, \mathsf{GenBits}, \mathsf{Verify})$ is *computationally binding* if the following property holds:

- **Computational binding:** There exists an efficient extractor $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$, where $\mathcal{E}_2$ is deterministic, and for all polynomials $\rho = \rho(\lambda)$, the following two properties hold:

– **CRS indistinguishability:** The following distributions are computationally indistinguishable:

$$\{\mathsf{Setup}(1^\lambda, 1^\rho)\} \overset{c}{\approx} \{(\mathsf{st}_\mathcal{E}, \widetilde{\mathsf{crs}}) \leftarrow \mathcal{E}_1(1^\lambda, 1^\rho) : \widetilde{\mathsf{crs}}\}.$$

– **Binding:** For all efficient adversaries $\mathcal{A}$, and sampling $(\mathsf{st}_\mathcal{E}, \widetilde{\mathsf{crs}}) \leftarrow \mathcal{E}_1(1^\lambda, 1^\rho)$ followed by $(\sigma^*, i^*, r^*, \pi^*) \leftarrow \mathcal{A}(1^\lambda, 1^\rho, \widetilde{\mathsf{crs}})$ and $r \leftarrow \mathcal{E}_2(\mathsf{st}_\mathcal{E}, \sigma^*)$, we have that

$$\Pr[r_{i^*} \neq r^* \wedge \mathsf{Verify}(\widetilde{\mathsf{crs}}, \sigma^*, i^*, r^*, \pi^*) = 1] = \mathsf{negl}(\lambda).$$

Pairing groups. In this section, we work in (asymmetric) pairing groups. We review the notion of a pairing below, as well as the kernel k-linear (k-KerLin) assumption [MRV15, KW15], which can be viewed as a *search* version of the k-linear assumption. As shown in [KW15], the k-KerLin assumption is weaker than the k-Lin assumption (in particular, k-Lin implies k-KerLin), but stronger than the CDH assumption.

Definition 4.11 (Prime-Order Pairing-Group Generator)**.** A prime-order (asymmetric) pairing group generator algorithm PairingGroupGen is an efficient algorithm that on input the security parameter 1^λ outputs a description $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ of two base groups $\mathbb{G}_1$ (generated by g_1), $\mathbb{G}_2$ (generated by g_2), and a target group $\mathbb{G}_T$, all of prime order p, together with an efficiently-computable mapping $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ (called the "pairing"). Finally, the mapping e is bilinear: for all $x, y \in \mathbb{Z}_p$, $e(g_1^x, g_2^y) = e(g_1, g_2)^{xy}$.

Definition 4.12 (Kernel k-Linear Assumption [MRV15, KW15])**.** The kernel k-linear (k-KerLin) assumption holds in $\mathbb{G}_2$ relative to a pairing-group generator PairingGroupGen if for all efficient adversaries $\mathcal{A}$, and sampling $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathsf{PairingGroupGen}(1^\lambda)$, $\mathbf{a} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$, and defining $\mathbf{A} \in \mathbb{Z}_p^{(k+1)\times k}$ as in Eq. (4.1), the following holds:

$$\Pr[g_1^{\mathbf{c}^\mathsf{T}} \leftarrow \mathcal{A}(\mathcal{G}, g_2^\mathbf{A}) : \mathbf{c}^\mathsf{T}\mathbf{A} = \mathbf{0} \wedge \mathbf{c} \neq \mathbf{0}] = \mathsf{negl}(\lambda).$$

We can define an analogous assumption over $\mathbb{G}_1$ (by interchanging the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ above).

Notation. For a matrix $\mathbf{A}$, we continue to write $g_1^\mathbf{A}$ and $g_2^\mathbf{A}$ to denote matrices of group elements (over $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively). In addition, if we have two matrices $\mathbf{A} \in \mathbb{Z}^{m\times\ell}$ and $\mathbf{B} \in \mathbb{Z}^{\ell\times n}$, we write $e(g_1^\mathbf{A}, g_2^\mathbf{B})$ to denote the operation that outputs $e(g_1, g_2)^{\mathbf{AB}} \in \mathbb{G}_T^{m\times n}$. In particular, the $(i,j)^{\text{th}}$ entry of $e(g_1^\mathbf{A}, g_2^\mathbf{B})$ is computed as

$$\left[e(g_1^\mathbf{A}, g_2^\mathbf{B})\right]_{i,j} = \prod_{k\in[\ell]} e(g_1^{a_{i,k}}, g_2^{b_{k,j}}).$$

Construction 4.13 (Publicly-Verifiable Hidden-Bits Generator from Pairings)**.** Let PairingGroupGen be a prime-order bilinear group generator algorithm. We construct a publicly-verifiable hidden-bits generator (HBG) as follow:

- $\mathsf{Setup}(1^\lambda, 1^\rho) \to \mathsf{crs}$: The setup algorithm starts by sampling $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathsf{PairingGroupGen}(1^\lambda)$ and a hash function $H \xleftarrow{\mathrm{R}} \mathcal{H}$ where $\mathcal{H}$ is a family of hash functions with domain $\mathbb{G}_1$ and range $\{0,1\}$. Next, it samples a matrix $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(\rho+k)\times k}$, vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k \xleftarrow{\mathrm{R}}$

$\mathbb{Z}_p^{\rho+k}$, and verification components $\mathbf{a} \xleftarrow{R} \mathbb{Z}_p^{k+1}$, $\mathbf{B}_1, \ldots, \mathbf{B}_\rho \xleftarrow{R} \mathbb{Z}_p^{k\times(k+1)}$. In addition, it samples $\mathbf{d} \xleftarrow{R} \mathbb{Z}_p^k$, and constructs the matrix

$$\mathbf{D} = \begin{pmatrix} \mathrm{diag}(\mathbf{d}) \\ \hline \mathbf{1}^\mathsf{T} \end{pmatrix} \in \mathbb{Z}_p^{(k+1)\times k}. \tag{4.2}$$

It computes $\hat{\mathbf{a}}^\mathsf{T} \leftarrow \mathbf{a}^\mathsf{T}\mathbf{D} \in \mathbb{Z}_p^k$, and for each $i \in [\rho]$, it computes $\mathbf{Z}_i \leftarrow \mathbf{w}_i\mathbf{a}^\mathsf{T} + \mathbf{V}\mathbf{B}_i \in \mathbb{Z}_p^{(\rho+k)\times(k+1)}$ and $\hat{\mathbf{B}}_i \leftarrow \mathbf{B}_i\mathbf{D} \in \mathbb{Z}_p^{k\times k}$. It outputs

$$\mathsf{crs} = \left(\mathcal{G}, H, g_1^{\mathbf{V}}, g_2^{\hat{\mathbf{a}}^\mathsf{T}}, g_2^{\mathbf{D}}, \left\{g_1^{\mathbf{w}_i}, g_1^{\mathbf{Z}_i}, g_2^{\hat{\mathbf{B}}_i}\right\}_{i\in[\rho]}\right).$$

- $\mathsf{GenBits}(\mathsf{crs}) \to (\sigma, r, \{\pi_i\}_{i\in[k]})$: On input $\mathsf{crs} = \left(\mathcal{G}, H, g_1^{\mathbf{V}}, g_2^{\hat{\mathbf{a}}^\mathsf{T}}, g_2^{\mathbf{D}}, \left\{g_1^{\mathbf{w}_i}, g_1^{\mathbf{Z}_i}, g_2^{\hat{\mathbf{B}}_i}\right\}_{i\in[\rho]}\right)$, sample $\mathbf{y} \xleftarrow{R} \mathbb{Z}_p^{\rho+k}$, and compute for each $i \in [\rho]$,

$$g_1^{t_i} \leftarrow g_1^{\mathbf{y}^\mathsf{T}\mathbf{w}_i} \quad \text{and} \quad g_1^{\mathbf{u}_i^\mathsf{T}} \leftarrow g_1^{\mathbf{y}^\mathsf{T}\mathbf{Z}_i}.$$

  Next, let $\sigma = g_1^{\mathbf{y}^\mathsf{T}\mathbf{V}}$, and for each $i \in [\rho]$, set $r_i \leftarrow H(g_1^{t_i})$ and $\pi_i = (g_1^{t_i}, g_1^{\mathbf{u}_i^\mathsf{T}})$. Output σ, r, and $\{\pi_i\}_{i\in[\rho]}$.

- $\mathsf{Verify}(\mathsf{crs}, \sigma, i, r_i, \pi_i)$: On input $\mathsf{crs} = \left(\mathcal{G}, H, g_1^{\mathbf{V}}, g_2^{\hat{\mathbf{a}}^\mathsf{T}}, g_2^{\mathbf{D}}, \left\{g_1^{\mathbf{w}_i}, g_1^{\mathbf{Z}_i}, g_2^{\hat{\mathbf{B}}_i}\right\}_{i\in[\rho]}\right)$, $\sigma = g_1^{\mathbf{c}^\mathsf{T}}$, $i \in [\rho]$, $r_i \in \{0,1\}$, and $\pi_i = (g_1^{t_i}, g_1^{\mathbf{u}_i^\mathsf{T}})$, output 1 if

$$e(g_1^{t_i}, g_2^{\hat{\mathbf{a}}^\mathsf{T}}) \cdot e(g_1^{\mathbf{c}^\mathsf{T}}, g_2^{\hat{\mathbf{B}}_i}) = e(g_1^{\mathbf{u}_i^\mathsf{T}}, g_2^{\mathbf{D}}) \tag{4.3}$$

  and $r_i = H(g_1^{t_i})$. If either check fails, output 0.

Correctness and security analysis. We now state the correctness and security theorems for Construction 4.13 and provide the proofs in Appendix C.2.

Theorem 4.14 (Correctness)**.** *Construction 4.13 is correct.*

Theorem 4.15 (Succinctness)**.** *Construction 4.13 is succinct.*

Theorem 4.16 (Computational Binding)**.** *Suppose* PairingGroupGen *outputs groups* $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ *such that the* k-Lin *assumption holds in* $\mathbb{G}_1$ *and the* k-KerLin *assumption holds in* $\mathbb{G}_2$*. Then, Construction 4.13 satisfies computational binding in binding mode.*

Theorem 4.17 (Statistical Simulation)**.** *If* $\mathcal{H}$ *satisfies statistical uniformity, then Construction 4.13 satisfies statistical simulation.*

4.3 Dual-Mode Hidden-Bits Generator with Malicious Security from k-Lin

We now show how to modify the k-Lin construction from Section 4.1 (Construction 4.3) to obtain a hidden-bits generator with security against malicious verifiers. Combined with Construction 3.4, this yields a dual-mode MDV-NIZK (Theorem 3.11). We refer to Section 1.2 for a high-level description of our approach.

Construction 4.18 (Dual-Mode HBG with Malicious Security from k-Lin)**.** Let ρ be the output length of the hidden-bits generator. We require the following primitives:

- Let $\mathsf{GroupGen}$ be a prime-order group generator algorithm.
- Let $\ell = 3\rho\lambda$ and define $\mathcal{T}_{\lambda,\ell} := \{S \subseteq [\ell] : |S| = \lambda\}$ to be the set of all subsets of $[\ell]$ that contains exactly λ elements. Let $G\colon \{0,1\}^\kappa \to \mathcal{T}_{\lambda,\ell}^\rho \times \mathbb{Z}_p^{\rho\ell}$ be a PRG with seed length $\kappa = \kappa(\lambda)$. Here, p is the order of the group $\mathbb{G}$ output by $\mathsf{GroupGen}$ (on input 1^λ).

  Constructing the PRG G. It is straightforward to construct a PRG with outputs in $\mathcal{T}_{\lambda,\ell}^\rho \times \mathbb{Z}_p^{\rho\ell}$ from a PRG with outputs in $\{0,1\}^{\rho\lambda\ell(1+\lceil\log p\rceil)}$. To see this, it suffices to give an efficient algorithm that maps from the uniform distribution on $\{0,1\}^{\lambda\ell(1+\lceil\log p\rceil)}$ to a distribution that is statistically close to uniform over $\mathcal{T}_{\lambda,\ell} \times \mathbb{Z}_p^\ell$. Take a string $\gamma \in \{0,1\}^{\lambda\ell(1+\lceil\log p\rceil)}$.

  - The first $\lambda\ell$ bits of γ are interpreted as ℓ blocks of λ-bit indices $i_1, \ldots, i_\ell \in \{0,1\}^\lambda$. These indices specify the set $S \subseteq \mathcal{T}_{\lambda,\ell}$ as follows. First, take $S_0 \leftarrow [\ell]$. For each $j \in [\lambda]$, take s_j to be the $(i_j \bmod |S_{j-1}|)^{\text{th}}$ element of S_{j-1} and define $S_j \leftarrow S_{j-1} \setminus \{s_j\}$. Define $S \leftarrow \{s_1, \ldots, s_\ell\} \in \mathcal{T}_{\lambda,\ell}$.
  - The remaining $\lambda\ell\lceil\log p\rceil$ bits of γ are taken to be the binary representation of an ℓ-dimensionalvector $\boldsymbol{\alpha} \in \mathbb{Z}^\ell$, where each component is a $\lambda\lceil\log p\rceil$-bit integer.

  The string $\gamma \in \{0,1\}^{\lambda\ell(1+\lceil\log p\rceil)}$ is mapped onto $(S, \boldsymbol{\alpha} \bmod p) \in \mathcal{T}_{\lambda,\ell} \times \mathbb{Z}_p^\ell$. By construction, this procedure maps from the uniform distribution over $\{0,1\}^{\lambda\ell(1+\lceil\log p\rceil}$ to a distribution that is statistically uniform over $\mathcal{T}_{\lambda,\ell} \times \mathbb{Z}_p^\ell$.

We construct the dual-mode designated-verifier hidden-bits generator with malicious security as follows:

- $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode}) \to \mathsf{crs}$: Let $\ell' = \rho\ell$. Sample $\mathcal{G} = (\mathbb{G}, p, g) \leftarrow \mathsf{GroupGen}(1^\lambda)$ and $H \xleftarrow{\text{R}} \mathcal{H}$, where $\mathcal{H}$ is a family of hash functions with domain $\mathbb{G}$ and range $\{0,1\}$. Next, it samples $\mathbf{V} \xleftarrow{\text{R}} \mathbb{Z}_p^{(\ell'+k)\times k}$ and vectors $\mathbf{w}_1, \ldots, \mathbf{w}_{\ell'} \in \mathbb{Z}_p^{\ell'+k}$ as follows:

  - If $\mathsf{mode} = \mathsf{hiding}$, sample $\mathbf{w}_i \xleftarrow{\text{R}} \mathbb{Z}_p^{\ell'+k}$ for all $i \in [\ell']$.
  - If $\mathsf{mode} = \mathsf{binding}$, sample $\mathbf{s}_i \xleftarrow{\text{R}} \mathbb{Z}_p^k$ and set $\mathbf{w}_i \leftarrow \mathbf{V}\mathbf{s}_i$ for all $i \in [\ell']$.

  Output $\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_{\ell'}})$.

- $\mathsf{KeyGen}(\mathsf{crs}) \to (\mathsf{pk}, \mathsf{sk})$: On input $\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_{\ell'}})$, sample $a \xleftarrow{\text{R}} \mathbb{Z}_p$ and $\mathbf{b}_1, \ldots, \mathbf{b}_{\ell'} \xleftarrow{\text{R}} \mathbb{Z}_p^k$. For each $i \in [\ell']$, compute $\mathbf{z}_i \leftarrow \mathbf{w}_i a + \mathbf{V}\mathbf{b}_i \in \mathbb{Z}_p^{\ell'+k}$ and output

$$\mathsf{pk} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_{\ell'}}) \quad \text{and} \quad \mathsf{sk} = (a, \mathbf{b}_1, \ldots, \mathbf{b}_{\ell'}).$$

- $\mathsf{GenBits}(\mathsf{crs}, \mathsf{pk}) \to (\sigma, r, \{\pi_i\}_{i\in[\rho]})$: On input $\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_{\ell'}})$ and the public key $\mathsf{pk} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_{\ell'}})$, sample $\mathbf{y} \xleftarrow{\text{R}} \mathbb{Z}_p^{\ell'+k}$ and compute for each $i \in [\ell']$

$$g^{t_i} \leftarrow g^{\mathbf{y}^\top \mathbf{w}_i} \quad \text{and} \quad g^{u_i} \leftarrow g^{\mathbf{y}^\top \mathbf{z}_i}.$$

Next, sample a PRG seed $\mathsf{s} \xleftarrow{R} \{0,1\}^\kappa$ and compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$ where $\hat{S}_i \in \mathcal{T}_{\lambda,\ell}$ for all $i \in [\rho]$ and $\boldsymbol{\alpha} \in \mathbb{Z}_p^{\rho\ell}$. Compute the shifted sets $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$ for each $i \in [\rho]$. Finally, compute

$$r_i \leftarrow H\left(\prod_{j \in S_i} g^{\alpha_j t_j}\right) \text{ and } \quad \pi_i \leftarrow \{(j, g^{t_j}, g^{u_j})\}_{j \in S_i}.$$

Output $\sigma = (\mathsf{s}, g^{\mathbf{y}^\mathsf{T}\mathbf{V}})$, r, and $\{\pi_i\}_{i \in [\rho]}$.

- $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$: On input $\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_{\ell'}})$, $\mathsf{sk} = (a, \mathbf{b}_1, \ldots, \mathbf{b}_{\ell'})$, $\sigma = (\mathsf{s}, g^{\mathbf{c}^\mathsf{T}})$, $i \in [\rho]$, $r_i \in \{0,1\}$, and $\pi_i = \{(j, g^{t_j}, g^{u_j})\}_{j \in S}$ for an implicitly-defined set $S \subseteq [\rho\ell]$, the verification algorithm performs the following checks:
  - Compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$ and the shifted set $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$. It checks that $S = S_i$ and outputs 0 if not.
  - It checks that $g^{u_j} = (g^{t_j a})(g^{\mathbf{c}^\mathsf{T}\mathbf{b}_j})$ for all $j \in S$, and outputs 0 if not.
  - It checks that $r_i = H\big(\prod_{j \in S} g^{\alpha_j t_j}\big)$ and outputs 0 if not.

  If all checks pass, the verification algorithm outputs 1.

Correctness and security analysis. We now state the correctness and security theorems for Construction 4.18 and provide the proofs in Appendix C.3.

Theorem 4.19 (Correctness)**.** *Construction 4.18 is correct.*

Theorem 4.20 (Succinctness)**.** *Construction 4.18 is succinct.*

Theorem 4.21 (CRS Indistinguishability)**.** *Suppose the k-Lin assumption holds for* GroupGen*. Then, Construction 4.18 satisfies CRS indistinguishability.*

Theorem 4.22 (Statistical Binding in Binding Mode)**.** *Construction 4.18 satisfies statistical binding in binding mode.*

Theorem 4.23 (Statistical Simulation in Hiding Mode)**.** *If G is a secure PRG and $\mathcal{H}$ satisfies statistical uniformity, then Construction 4.18 satisfies statistical simulation in hiding mode against malicious verifiers.*

5 Dual-Mode Hidden-Bits Generators from QR

In this section, we show how to similarly construct dual-mode hidden-bits generators from subgroup-indistinguishability-type assumptions [BG10], and specifically, from the QR and DCR assumptions. We begin in Section 5.1 with a basic construction from QR (the analog of Construction 4.3), and then show in Section 5.2 how we can use similar techniques from Section 4.3 to obtain a dual-mode hidden-bits generator with malicious security also from the QR assumption. While the construction with malicious security (Construction 5.9) subsumes the basic construction (Construction 5.3), we begin with the basic construction because it is both simpler to understand and contains all of the essential ingredients for realizing HBGs from a subgroup-indistinguishability-type assumption. Finally, in Appendix E, we show how to adapt our techniques to also obtain dual-mode HBGs from the DCR assumption.

5.1 Dual-Mode Hidden-Bits Generator from QR

In this section, we describe our construction of a dual-mode hidden-bits generator from the quadratic residuosity (QR) assumption [GM82]. We first recall some basic notation and the QR assumption.

Notation. Our construction works with a modulus $N = pq$ that is a product of safe primes p, q: namely, $p = 2p' + 1$ and $q = 2q' + 1$ for primes p' and q'. We write $\mathbb{J}_N$ to denote the multiplicative subgroup of $\mathbb{Z}_N^*$ with Jacobi symbol $+1$ (which has order $2p'q'$) and $\mathbb{QR}_N$ to denote the multiplicative subgroup of quadratic residues modulo N (which has order $p'q'$). In the following description, we use the fact that $\mathbb{J}_N$ splits into two subgroups of coprime order: namely, $\mathbb{J}_N = \mathbb{QR}_N \times \mathbb{H}$, where $\mathbb{H}$ is the multiplicative subgroup $\{\pm 1\}$ of order 2 generated by -1. When N is a product of safe primes, $\mathbb{QR}_N$ is also *cyclic*. As such, we will write g to denote a generator of $\mathbb{QR}_N$ and $h = -1$ to denote the generator of $\mathbb{H}$. We will typically write elements of $\mathbb{J}_N$ with a bar (e.g., $\overline{c}, \overline{t}$). In the analysis, it will be more conducive to analyze the components in their respective subgroups. In this case, we will often express elements of $\mathbb{J}_N$ as $\overline{c} = g^c h^{\hat{c}} \in \mathbb{J}_N$, for exponents $c \in \mathbb{Z}_{p'q'}$ and $\hat{c} \in \mathbb{Z}_2$.

Definition 5.1 (Sampling Safe Prime Product Modulus)**.** A safe prime product modulus sampler $\mathsf{SampleModulus}$ is an efficient algorithm that on input the security parameter λ, outputs a tuple (N, p, q) where $N = pq$, $p = 2p' + 1$, $q = 2q' + 1$, for distinct primes p', q' where $1/p', 1/q' = \mathsf{negl}(\lambda)$.

Definition 5.2 (Quadratic Residuosity Assumption [GM82])**.** A safe prime product sampler $\mathsf{SampleModulus}$ satisfies the quadratic residuosity (QR) assumption if, for any efficient adversary $\mathcal{A}$, and any sampling $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$, $x \xleftarrow{R} \mathbb{QR}_N$, where $p = 2p' + 1$ and $q = 2q' + 1$,

$$|\Pr[\mathcal{A}(N, x) = 1] - \Pr[\mathcal{A}(N, (-1) \cdot x) = 1]| = \mathsf{negl}(\lambda).$$

Our QR-based construction is conceptually similar to the QR-based trapdoor hash function of [DGI+19, Section 4.3]. Like [DGI+19], it uses the predicate $\mathsf{LEQ} \colon \mathbb{J}_N \times \mathbb{J}_N \to \{0, 1\}$ that outputs 1 if the bit representation of its first argument is no larger than that of its second argument in some lexicographical ordering. In particular, for all $x_1, x_2 \in \mathbb{J}_N$, it will be the case that $\mathsf{LEQ}(x_1, x_2) = 1 - \mathsf{LEQ}(x_2, x_1)$.

Construction 5.3 (Dual-Mode Hidden-Bits Generator from QR)**.** Let ρ be the output length of the hidden-bits generator. Our QR-based dual-mode hidden-bits generator (HBG) works as follows:

- $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode}) \to \mathsf{crs}$: Sample $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$, and let g be a generator of $\mathbb{QR}_N$. Let $h = -1$ be the generator of $\mathbb{H} = \{\pm 1\}$. The setup algorithm then samples a vector $\mathbf{v} \xleftarrow{R} \mathbb{Z}^\rho_{\lfloor N/2 \rfloor}$, scalars $s_1, \ldots, s_\rho \xleftarrow{R} \mathbb{Z}_{\lfloor N/2 \rfloor}$, and sets $\hat{\mathbf{w}}_i \in \mathbb{Z}_2^\rho$ for $i \in [\rho]$ as follows:
  - If $\mathsf{mode} = \mathsf{hiding}$, set $\hat{\mathbf{w}}_i \leftarrow \mathbf{e}_i$ where $\mathbf{e}_i \in \mathbb{Z}_2^\rho$ is the i^{th} basis vector.
  - If $\mathsf{mode} = \mathsf{binding}$, set $\hat{\mathbf{w}}_i \leftarrow \mathbf{0}$.

  Finally, it sample a hash function $H \xleftarrow{R} \mathcal{H}$, where $\mathcal{H}$ is a family of hash functions with domain $\mathbb{Z}_N$ and range $\{0, 1\}^\lambda$. Output $\mathsf{crs} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\hat{\mathbf{w}}_1}, \ldots, g^{s_\rho \mathbf{v}} h^{\hat{\mathbf{w}}_\rho})$.
- $\mathsf{KeyGen}(\mathsf{crs}) \to (\mathsf{pk}, \mathsf{sk})$: On input $\mathsf{crs} = (N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_\rho)$, sample $a_\tau, b_{\tau,i} \xleftarrow{R} \mathbb{Z}_{\lfloor N/2 \rfloor}$ for all $\tau \in [T]$ and $i \in [\rho]$, where $T = 2(\lambda + \lceil \log N \rceil)$. It computes and outputs the public key $\mathsf{pk} = \{\overline{\mathbf{v}}^{b_{\tau,i}} \overline{\mathbf{w}}_i^{a_\tau}\}_{\tau \in [T], i \in [\rho]}$ and $\mathsf{sk} = \{a_\tau, b_{\tau,i}\}_{\tau \in [T], i \in [\rho]}$.

- $\mathsf{GenBits}(\mathsf{crs}, \mathsf{pk}) \to (\sigma, r, \{\pi_i\}_{i\in[\rho]})$: On input $\mathsf{crs} = (N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_\rho)$ and $\mathsf{pk} = \{\overline{\mathbf{z}}_{\tau,i}\}_{\tau\in[T],i\in[\rho]}$, sample $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{\lfloor N/2\rfloor}$ and compute for all $\tau \in [T]$ and $i \in [\rho]$

$$\bar{c} \leftarrow \prod_{j\in[\rho]} \overline{v}_j^{y_i} \quad \text{and} \quad \bar{t}_i \leftarrow \prod_{j\in[\rho]} \overline{w}_{i,j}^{y_j} \quad \text{and} \quad \overline{u}_{\tau,i} \leftarrow \prod_{j\in[\rho]} \overline{z}_{\tau,i,j}^{y_j}.$$

  For each $i \in [\rho]$, let $u_i \leftarrow H(\overline{u}_{1,i}, \ldots, \overline{u}_{T,i})$ and $r_i \leftarrow \mathsf{LEQ}(\bar{t}_i, \bar{t}_i h)$. Output $\sigma = \bar{c}$, r, and $\pi = \{(\bar{t}_i, u_i)\}_{i\in[\rho]}$.

- $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$: On input $\mathsf{crs} = (N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_\rho)$, the verification key $\mathsf{sk} = \{a_\tau, b_{\tau,i}\}_{\tau\in[T],i\in[\rho]}$, $\sigma = \bar{c}$, $i \in [\rho]$, $r_i \in \{0, 1\}$, and $\pi_i = (\bar{t}_i, u_i)$, output 1 if

$$u_i = H\big(\bar{t}_i^{a_1}\bar{c}^{b_{1,i}}, \ldots, \bar{t}_i^{a_T}\bar{c}^{b_{T,i}}\big), \tag{5.1}$$

  and $r_i = \mathsf{LEQ}(\bar{t}_i, \bar{t}_i h)$. Otherwise, output 0.

The group elements that are input to algorithms KeyGen, $\mathsf{GenBits}$, and Verify are assumed to be elements of the subgroup $\mathbb{J}_N$ of $\mathbb{Z}_N^*$. Note that since membership in $\mathbb{J}_N$ can be efficiently tested given the modulus N (by computing the Jacobi symbol), each of the algorithms will first check this condition, and proceed *only if* all of the inputs are from the correct domains. We omit this explicit check in the above description for ease of exposition.

Correctness and security analysis. We now state the correctness and security theorems for Construction 5.3 and give the proofs in Appendix D.1.

Theorem 5.4 (Correctness)**.** *Construction 5.3 is correct.*

Theorem 5.5 (Succinctness)**.** *Construction 5.3 is succinct.*

Theorem 5.6 (CRS Indistinguishability)**.** *Suppose the* QR *assumption holds with respect to* $\mathsf{SampleModulus}$*. Then, Construction 5.3 satisfies CRS indistinguishability.*

Theorem 5.7 (Statistical Binding in Binding Mode)**.** *If* $\mathcal{H}$ *is pairwise independent, Construction 5.3 satisfies statistical binding in binding mode.*

Theorem 5.8 (Statistical Simulation in Hiding Mode)**.** *Construction 5.3 satisfies statistical simulation in hiding mode.*

5.2 Dual-Mode Hidden-Bits Generator with Malicious Security from QR

In this section, we show how to extend Construction 5.3 to obtain a dual-mode hidden-bits generator with malicious security from the QR assumption. Our construction is conceptually similar to Construction 4.18 from k-Lin.

Construction 5.9 (Dual-Mode HBG with Malicious Security from QR)**.** Let ρ be the output length of the hidden-bits generator. We rely on a similar set of building blocks as Construction 4.18:

- Let $\mathsf{SampleModulus}$ be a safe prime modulus sampler. Let $T = 2(\lambda + \lceil \log N \rceil)$, where $\lceil \log N \rceil$ is a bound on the bit-length of the modulus output by $\mathsf{SampleModulus}$ (on input 1^λ).

- Let $\ell = 36\rho\lambda^2 T$ and define $\mathcal{T}_{\lambda,\ell} := \{S \subseteq [\ell] : |S| = \lambda\}$ to be the set of all subsets of $[\ell]$ that contains exactly λ elements. Let $G\colon \{0,1\}^\kappa \to \mathcal{T}_{\lambda,\ell}^\rho \times \mathbb{Z}_2^{\rho\ell}$ be a PRG with seed length $\kappa = \kappa(\lambda)$. Here, N is the modulus output by $\mathsf{SampleModulus}$. We refer to Construction 4.18 for a description of how to construct such a PRG.

We construct the dual-mode designated-verifier HBG with malicious security as follows:

- $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode}) \to \mathsf{crs}$: Let $\ell' = \rho\ell$. Sample $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$. Let g be a generator of $\mathbb{QR}_N$ and $h = -1$ be the generator of $\mathbb{H} = \{\pm 1\}$. The setup algorithm samples a vector $\mathbf{v} \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N/2 \rfloor}^{\ell'}$, scalars $s_1, \ldots, s_{\ell'} \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N/2 \rfloor}$, and sets $\hat{\mathbf{w}}_i \in \mathbb{Z}_2^{\ell'}$ for $i \in [\ell']$ as follows:

  - If $\mathsf{mode} = \mathsf{hiding}$, set $\hat{\mathbf{w}}_i \leftarrow \mathbf{e}_i$ where $\mathbf{e}_i \in \mathbb{Z}_2^{\ell'}$ is the i^{th} basis vector.
  - If $\mathsf{mode} = \mathsf{binding}$, set $\hat{\mathbf{w}}_i \leftarrow \mathbf{0}$.

  Finally, it sample a hash function $H \xleftarrow{\mathrm{R}} \mathcal{H}$, where $\mathcal{H}$ is a family of hash functions with domain $\mathbb{Z}_N$ and range $\{0,1\}^\lambda$. Output $\mathsf{crs} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\hat{\mathbf{w}}_1}, \ldots, g^{s_{\ell'} \mathbf{v}} h^{\hat{\mathbf{w}}_{\ell'}})$.

- $\mathsf{KeyGen}(\mathsf{crs}) \to (\mathsf{pk}, \mathsf{sk})$: On input $\mathsf{crs} = (N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_{\ell'})$, sample $a_\tau, b_{\tau,i} \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N/2 \rfloor}$ for all $\tau \in [T]$ and $i \in [\ell']$. It computes and outputs the public key $\mathsf{pk} = \{\overline{\mathbf{v}}^{b_{\tau,i}} \overline{\mathbf{w}}_i^{a_\tau}\}_{\tau \in [T], i \in [\ell']}$ and $\mathsf{sk} = \{a_\tau, b_{\tau,i}\}_{\tau \in [T], i \in [\ell']}$.

- $\mathsf{GenBits}(\mathsf{crs}, \mathsf{pk}) \to (\sigma, r, \{\pi_i\}_{i \in [\rho]})$: On input $\mathsf{crs} = (N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_{\ell'})$ and $\mathsf{pk} = \{\overline{\mathbf{z}}_{\tau,i}\}_{\tau \in [T], i \in [\ell']}$, sample $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N/2 \rfloor}^{\ell'}$ and compute for all $\tau \in [T]$ and $i \in [\ell']$

$$\overline{c} \leftarrow \prod_{j \in [\ell']} \overline{v}_j^{y_i} \quad \text{and} \quad \overline{t}_i \leftarrow \prod_{j \in [\ell']} \overline{w}_{i,j}^{y_j} \quad \text{and} \quad \overline{u}_{\tau,i} \leftarrow \prod_{j \in [\ell']} \overline{z}_{\tau,i,j}^{y_j}.$$

  In addition, for each $i \in [\ell']$, compute $u_i \leftarrow H(\overline{u}_{1,i}, \ldots, \overline{u}_{T,i})$. Next, sample a PRG seed $\mathsf{s} \xleftarrow{\mathrm{R}} \{0,1\}^\kappa$ and compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$ where $\hat{S}_i \in \mathcal{T}_{\lambda,\ell}$ for all $i \in [\rho]$ and $\boldsymbol{\alpha} \in \mathbb{Z}_2^{\rho\ell}$. For each $i \in [\rho]$, compute the shifted sets $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$ and set

$$\overline{t}'_i \leftarrow \prod_{j \in S_i} \overline{t}_j^{\alpha_j} \quad \text{and} \quad r_i \leftarrow \mathsf{LEQ}(\overline{t}'_i, \overline{t}'_i h) \quad \text{and} \quad \pi_i \leftarrow \{(j, \overline{t}_j, u_j)\}_{j \in S_i}.$$

  Output $\sigma = (\mathsf{s}, \overline{c})$, r, and $\pi = \{\pi_i\}_{i \in [\rho]}$.

- $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$: On input $\mathsf{crs} = (N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_{\ell'})$, $\mathsf{sk} = \{a_\tau, b_{\tau,i}\}_{\tau \in [T], i \in [\ell']}$, $\sigma = (\mathsf{s}, \overline{c})$, $i \in [\rho]$, $r_i \in \{0,1\}$, and $\pi_i = \{(h, \overline{t}_j, u_j)\}_{j \in S}$ for an implicitly-defined set $S \subseteq [\ell']$, the verification algorithm performs the following checks:

  - Compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$ and the shifted set $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$. It checks that $S = S_i$ and outputs 0 if not.
  - It checks that

$$u_j = H\big(\overline{t}_j^{a_1} \overline{c}^{b_{1,j}}, \ldots, \overline{t}_j^{a_T} \overline{c}^{b_{T,j}}\big), \tag{5.2}$$

    for all $j \in S$, and outputs 0 if not.
  - It computes $\overline{t}'_i \leftarrow \prod_{j \in S} \overline{t}_j^{\alpha_j}$ and checks that $r_i = \mathsf{LEQ}(\overline{t}'_i, \overline{t}'_i h)$.

The group elements that are input to algorithms KeyGen, GenBits, and Verify are assumed to be elements of the subgroup $\mathbb{J}_N$ of $\mathbb{Z}_N^*$. Note that since membership in $\mathbb{J}_N$ can be efficiently tested given the modulus N (by computing the Jacobi symbol), each of the algorithms will first check this condition, and proceed *only if* all of the inputs are from the correct domains. We omit this explicit check in the above description for ease of exposition.

Correctness and security analysis. We now state the correctness and security theorems for Construction 5.9 and give the proofs in Appendix D.2.

Theorem 5.10 (Correctness)**.** *Construction 5.9 is correct.*

Theorem 5.11 (Succinctness)**.** *Construction 5.9 is succinct.*

Theorem 5.12 (CRS Indistinguishability)**.** *Suppose the* QR *assumption holds with respect to* SampleModulus*. Then, Construction 5.9 satisfies CRS indistinguishability.*

Theorem 5.13 (Statistical Binding in Binding Mode)**.** *If $\mathcal{H}$ is pairwise independent, Construction 5.9 satisfies statistical binding in binding mode.*

Theorem 5.14 (Statistical Simulation in Hiding Mode)**.** *Construction 5.9 satisfies statistical simulation in hiding mode.*

6 Instantiations and Extensions

In this section, we provide the main implications of our framework for constructing statistical (and more generally, dual-mode) NIZKs. We conclude by describing two simple extensions to augment our NIZKs with additional properties.

Dual-mode MDV-NIZKs. By instantiating Construction 3.4 with a dual-mode MDV hidden-bits generator (e.g., Constructions 4.18, 5.9 and E.16), we obtain a dual-mode MDV-NIZK (Theorems 3.5, 3.7 and 3.11). We summarize our instantiations with the following corollaries:

Corollary 6.1 (Dual-Mode MDV-NIZK from k-Lin)**.** *Under the k-*Lin *assumption over pairing-free groups (for any $k \geq 1$), there exists a statistical MDV-NIZK argument (with non-adaptive soundness) in the common* random *string model, and a computational MDV-NIZK proof (with adaptive soundness) for* NP *in the common* reference *string model.*

Corollary 6.2 (Dual-Mode MDV-NIZK from QR or DCR)**.** *Under the* QR *or* DCR *assumptions, there exists a statistical MDV-NIZK argument (with non-adaptive soundness) and a computational MDV-NIZK proof (with adaptive soundness) for* NP *in the common reference string model.*

Remark 6.3 (Perfect Zero-Knowledge DV-NIZK from k-Lin)**.** As discussed in Remark C.8, Construction 4.3 satisfies *perfect* statistical simulation if the hash function $H\colon \mathbb{G} \setminus \{g^0\} \to \{0,1\}$ is perfectly uniform and there is an efficient algorithm to exactly sample $t \xleftarrow{R} \mathbb{Z}_p$ such that $H(g^t) = r$ for any $r \in \{0,1\}$. Here we describe a straightforward candidate for H when $\mathbb{G} = E(\mathbb{F}_p)$ is an elliptic-curve group of prime-order. We can write $E(\mathbb{F}_p) = \{y^2 = x^3 + Ax + B \mid (x,y) \in \mathbb{F}_p^2\} \cup \{\mathcal{O}\}$, where $\mathcal{O}$ is the identity element. For a point $(x,y) \in \mathbb{G} \setminus \{\mathcal{O}\}$, we define the hash function $H(x,y) := \mathrm{sign}(y)$, where $\mathrm{sign}(y)$ outputs 0 if $y \in \mathbb{F}_p$ is lexicographically smaller than $-y \in \mathbb{F}_p$, and 1 otherwise. Since

$(x, y) \in \mathbb{G}$ implies that $(x, -y) \in \mathbb{G}$, H perfectly partitions $\mathbb{G} \setminus \{\mathcal{O}\}$ into two equal-size sets, and correspondingly, H is perfectly uniform. Next, it is straightforward to sample $t_i \xleftarrow{R} \mathbb{G} \setminus \{\mathcal{O}\}$ such that $H(t) = r$. Simply sample $(x, y) \xleftarrow{R} \mathbb{G} \setminus \{\mathcal{O}\}$, and output either (x, y) or $(x, -y)$, depending on r. Thus, $\mathcal{H}$ satisfies the required properties and combined with Construction 4.3, gives a hidden-bits generator with perfect simulation in hiding mode. Combined with Construction 3.4, we obtain a perfect DV-NIZK argument from the k-Lin assumption in pairing-free groups. Note that our MDV-NIZK construction from the k-Lin assumption (Construction 4.18) does not satisfy perfect simulation in hiding mode, so we do not obtain a perfect MDV-NIZK argument.

Publicly-verifiable statistical NIZK arguments. By instantiating Construction B.1 with a publicly-verifiable hidden-bits generator satisfying statistical simulation (e.g., Construction 4.13), we obtain a publicly-verifiable statistical NIZK argument in the common reference string model:

Corollary 6.4 (Publicly-Verifiable Statistical NIZK Argument from Pairings)**.** *Suppose that the k-Lin assumption holds in $\mathbb{G}_1$ and the k-KerLin assumption holds in $\mathbb{G}_2$ (for any $k \geq 1$) over a pairing group. Then, there exists a publicly-verifiable statistical NIZK argument for* NP *(with non-adaptive soundness) in the common reference string model.*

Extensions. We conclude by describing two simple extensions to our framework to support languages where statements can have an arbitrary a priori unbounded polynomial length as well as how to obtain a proof of knowledge.

Remark 6.5 (Unbounded Statement Size)**.** All of our NIZK constructions in this paper assumed an a priori bound on the length n of the statements in the language (which determines the length of the hidden-bits string we require), and the length of the CRS scales with n. Using the same idea from [QRW19, §7], it is straightforward to adapt the construction to have a fixed-size CRS (independent of n) that supports proving statements of arbitrary $\mathsf{poly}(\lambda)$ size. We require a non-interactive commitment in the CRS model, and we will use 3-SAT as our underlying NP-complete language. To prove that a specific 3-CNF formula is satisfiable, the prover would commit to the satisfying assignment (one variable at a time), and then use the NIZK to prove that the committed values indeed satisfy each clause. Observe that we now only require a NIZK that supports a *fixed-size* language to implement this transformation. Moreover, if we instantiate the commitment with a "dual-mode" commitment, where in one mode, the commitment is statistically binding, and in the other, is statistically hiding, then we retain the dual-mode properties of the underlying NIZK. We can build dual-mode commitments from any lossy public-key encryption scheme [BHY09] (implied by standard intractability assumptions like DDH, QR, and DCR). Specifically, the CRS would contain a public key for the encryption scheme, and a commitment to a bit $b \in \{0, 1\}$ would be an encryption of b with fresh randomness r. The randomness r then serves as the commitment opening. When the lossy encryption scheme is injective, then the commitment is statistically binding and if the encryption scheme is lossy, then the commitment scheme is statistically hiding.

Remark 6.6 (Proofs of Knowledge)**.** It is also straightforward to update our NIZK constructions to obtain a proof of knowledge via the classic transformation where the prover encrypts the witness (using a public-key encryption scheme) and gives a proof that the encrypted witness satisfies the relation [DP92]. To preserve the dual-mode properties of the NIZK, we again require a lossy public-key encryption scheme [BHY09] (similar to Remark 6.5).

Acknowledgments

We thanks the anonymous Eurocrypt reviewers for helpful feedback on this work.

References

[AF07] Masayuki Abe and Serge Fehr. Perfect NIZK with adaptive soundness. In *TCC*, 2007.

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *EUROCRYPT*, 2004.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO*, 2004.

[BCG+19] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In *CRYPTO*, 2019.

[BCGI18] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. Compressing vector OLE. In *ACM CCS*, 2018.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, 1988.

[BG10] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *CRYPTO*, 2010.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 1–35. Springer, 2009.

[Blu86] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, volume 1, 1986.

[BY92] Mihir Bellare and Moti Yung. Certifying cryptographic tools: The case of trapdoor permutations. In *CRYPTO*, 1992.

[CCH+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In *STOC*, 2019.

[CDI+19] Melissa Chase, Yevgeniy Dodis, Yuval Ishai, Daniel Kraschewski, Tianren Liu, Rafail Ostrovsky, and Vinod Vaikuntanathan. Reusable non-interactive secure computation. In *CRYPTO*, 2019.

[CH19] Geoffroy Couteau and Dennis Hofheinz. Designated-verifier pseudorandom generators, and their applications. In *EUROCRYPT*, 2019.

[CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, 2003.

[CKS08] David Cash, Eike Kiltz, and Victor Shoup. The twin Diffie-Hellman problem and applications. In *EUROCRYPT*, 2008.

[CL18] Ran Canetti and Amit Lichtenberg. Certifying trapdoor permutations, revisited. In *TCC*, 2018.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, 1998.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, 2002.

[DDO+01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *CRYPTO*, 2001.

[DFN06] Ivan Damgård, Nelly Fazio, and Antonio Nicolosi. Non-interactive zero-knowledge from homomorphic encryption. In *TCC*, 2006.

[DGI+19] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications. In *CRYPTO*, 2019.

[DMP87] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In *CRYPTO*, 1987.

[DP92] Alfredo De Santis and Giuseppe Persiano. Zero-knowledge proofs of knowledge without interaction (extended abstract). In *FOCS*, 1992.

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge L. Villar. An algebraic framework for Diffie-Hellman assumptions. In *CRYPTO*, 2013.

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *FOCS*, 1990.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1), 1999.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, 1986.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *STOC*, 1982.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1), 1989.

[Gol11] Oded Goldreich. Basing non-interactive zero-knowledge on (enhanced) trapdoor permutations: The state of the art. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*. 2011.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In *EUROCRYPT*, 2006.

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3), 2012.

[GR13] Oded Goldreich and Ron D. Rothblum. Enhancements of trapdoor permutations. *J. Cryptology*, 26(3), 2013.

[Gro10] Jens Groth. Short non-interactive zero-knowledge proofs. In *ASIACRYPT*, 2010.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4), 1999.

[HJO+16] Brett Hemenway, Zahra Jafargholi, Rafail Ostrovsky, Alessandra Scafuro, and Daniel Wichs. Adaptively secure garbled circuits from one-way functions. In *CRYPTO*, 2016.

[HJR16] Dennis Hofheinz, Tibor Jager, and Andy Rupp. Public-key encryption with simulation-based selective-opening security and compact ciphertexts. In *TCC*, 2016.

[HK07] Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO*, 2007.

[HU19] Dennis Hofheinz and Bogdan Ursu. Dual-mode NIZKs from obfuscation. In *ASIACRYPT*, 2019.

[KNYY19a] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Designated verifier/prover and preprocessing NIZKs from Diffie-Hellman assumptions. In *EUROCRYPT*, 2019.

[KNYY19b] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Exploring constructions of compact NIZKs from various assumptions. In *CRYPTO*, 2019.

[KW15] Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. In *EUROCRYPT*, 2015.

[KW18] Sam Kim and David J. Wu. Multi-theorem preprocessing NIZKs from lattices. In *CRYPTO*, 2018.

[LQR+19] Alex Lombardi, Willy Quach, Ron D. Rothblum, Daniel Wichs, and David J. Wu. New constructions of reusable designated-verifier NIZKs. In *CRYPTO*, 2019.

[MRV15] Paz Morillo, Carla Ràfols, and Jorge L. Villar. Matrix computational assumptions in multilinear groups. *IACR Cryptology ePrint Archive*, 2015.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, 1999.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In *CRYPTO*, 2019.

[PsV06] Rafael Pass, Abhi shelat, and Vinod Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In *CRYPTO*, 2006.

[QRW19] Willy Quach, Ron D. Rothblum, and Daniel Wichs. Reusable designated-verifier NIZKs for all NP from CDH. In *EUROCRYPT*, 2019.

[Sha07] Hovav Shacham. A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. *IACR Cryptology ePrint Archive*, 2007.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *STOC*, 2014.

[YYHK16] Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Adversary-dependent lossy trapdoor function from hardness of factoring semi-smooth RSA subgroup moduli. In *CRYPTO*, 2016.

A Analysis of Construction 3.4 (Dual-Mode DV-NIZK)

In this section, we provide complete proofs for statements in Section 3.1.

Proof of Theorem 3.7 (Statistical Soundness). The argument proceeds very similarly to the corresponding proof of adaptive statistical soundness from [QRW19]. We use a simple hybrid argument:

- Hyb_0: This is the real soundness experiment where the challenger begins by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$, where $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$, $\mathsf{pk} = \mathsf{pk}_{\mathsf{HBG}}$, and $\mathsf{sk} = \mathsf{sk}_{\mathsf{HBG}}$. The challenger gives crs and pk to $\mathcal{A}$. Adversary $\mathcal{A}$ can then make verification queries on pairs (x, π) and the challenger responds with $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x, \pi)$. At the end of the experiment, the adversary outputs (x^*, π^*) and the output of the experiment is 1 if $x^* \notin \mathcal{L}$ and $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x^*, \pi^*) = 1$.
- Hyb_1: Same as Hyb_0 except that, at the end of the experiment, after the adversary outputs its statement $x^* \in \{0,1\}^n$ and proof $\pi^* = \left(\sigma^*, I^*, r^*_{I^*}, \{\pi^*_{\mathsf{HBG},i}\}_{i \in I^*}, \pi^*_{\mathsf{HBM}}\right)$, the challenger performs the following additional check:
  - Compute $r \leftarrow \mathsf{Open}(\mathsf{crs}_{\mathsf{HBG}}, \sigma^*)$. If $\mathsf{HBG.Verify}(\mathsf{crs}_{\mathsf{HBG}}, \mathsf{sk}_{\mathsf{HBG}}, \sigma^*, i, r^*_i, \pi^*_{\mathsf{HBG},i}) = 1$ for all $i \in I^*$ and $r_{I^*} \neq r^*_{I^*}$, then the challenger aborts the experiments and outputs $\perp$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of an execution of experiment Hyb_i with adversary $\mathcal{A}$. We now show that $\mathsf{Hyb}_0(\mathcal{A}) \overset{s}{\approx} \mathsf{Hyb}_1(\mathcal{A})$ and that $\Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.

Lemma A.1. *If Π_{HBG} is statistically binding in binding mode, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \overset{s}{\approx} \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. The only difference between Hyb_0 and Hyb_1 is the additional check the challenger performs when computing $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x^*, \pi^*)$ which precisely coincides with the binding property of the hidden-bits generator. An adversary that causes Hyb_1 to output $\perp$ with noticeable probability can break the binding property with the same probability (formally, the reduction algorithm can simulate the $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \cdot, \cdot)$ queries via oracle access to $\mathsf{HBG.Verify}(\mathsf{crs}_{\mathsf{HBG}}, \mathsf{sk}_{\mathsf{HBG}}, \cdot, \cdot, \cdot, \cdot)$ in the binding game). □

Lemma A.2. *For all adversaries $\mathcal{A}$, $\Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.*

Proof. Let $x^* \in \{0,1\}^n$ and $\pi^* = (\sigma^*, I^*, r^*_{I^*}, \{\pi^*_{\mathsf{HBG},i}\}_{i \in I^*}, \pi^*_{\mathsf{HBM}})$ be the adversary's output in Hyb_1. For $\mathsf{Hyb}_1(\mathcal{A})$ to output 1, the following conditions must hold:

- $r_{I^*} = r^*_{I^*}$ where $r \leftarrow \mathsf{Open}(\mathsf{crs}_{\mathsf{HBG}}, \sigma^*)$;
- $\mathsf{HBM.Verify}(1^\lambda, I^*, r_{I^*} \oplus s_{I^*}, x^*, \pi^*_{\mathsf{HBM}}) = 1$.

Fix any commitment string $\sigma \in \{0,1\}^\ell$ and let $r \leftarrow \mathsf{Open}(\mathsf{crs}_{\mathsf{HBG}}, \sigma)$ be the associated sequence of bits. For a randomly and independently sampled $s \xleftarrow{R} \{0,1\}^\rho$, $r_{I^*} \oplus s_{I^*}$ is also uniformly random. By soundness of Π_{HBM},

$$\Pr[\mathsf{HBM.Verify}(1^\lambda, I^*, r_{I^*} \oplus s_{I^*}, x^*, \pi^*_{\mathsf{HBM}})] = \varepsilon(\lambda).$$

Taking a union bound over all possible commitments $\sigma \in \{0,1\}^\ell$, we have that

$$\Pr[\exists\, \sigma \in \{0,1\}^\ell : \mathsf{HBM.Verify}(1^\lambda, I^*, r_{I^*} \oplus s_{I^*}, x^*, \pi^*_{\mathsf{HBM}})] \leq 2^\ell \cdot \varepsilon(\lambda) = \mathsf{negl}(\lambda),$$

where in the above expression, $r \leftarrow \mathsf{Open}(\mathsf{crs}_{\mathsf{HBG}}, \sigma^*)$. Thus, $\mathsf{Hyb}_1(\mathcal{A})$ outputs 1 with negligible probability. □

Since the distributions $\mathsf{Hyb}_0(\mathcal{A})$ and $\mathsf{Hyb}_1(\mathcal{A})$ are statistically indistinguishable, the probability that $\mathsf{Hyb}_0(\mathcal{A})$ outputs 1 (i.e., $\mathcal{A}$ breaks soundness) is negligible. □

Proof of Theorem 3.11 (Statistical Zero-Knowledge). The proof follows by essentially the same argument as the proof of Theorem 3.8. Specifically, let $\mathcal{S}_{\mathsf{HBM}}$ be the zero-knowledge simulator for Π_{HBM} and $\mathcal{S}_{\mathsf{HBG}} = (\mathcal{S}_{\mathsf{HBG},1}, \mathcal{S}_{\mathsf{HBG},2})$ be the simulator for Π_{HBG} in hiding mode. We construct the the zero-knowledge simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ for Π_{dvNIZK} as follows:

- $\mathcal{S}_1(1^\lambda) \to (\mathsf{st}_{\mathcal{S}}, \widetilde{\mathsf{crs}})$: Run $(\mathsf{st}_{\mathsf{HBG}}, \widetilde{\mathsf{crs}}_{\mathsf{HBG}}) \leftarrow \mathcal{S}_{\mathsf{HBG}}(1^\lambda, 1^\rho)$ and sample $\widetilde{s} \xleftarrow{R} \{0,1\}^\rho$. Output $\mathsf{st}_{\mathcal{S}} = \mathsf{st}_{\mathsf{HBG}}$ and $\widetilde{\mathsf{crs}} = (\lambda, s, \widetilde{\mathsf{crs}}_{\mathsf{HBG}})$.
- $\mathcal{S}_2(\mathsf{st}_{\mathcal{S}}, \mathsf{pk}, x) \to \widetilde{\pi}$: On input $\mathsf{st}_{\mathcal{S}} = \mathsf{st}_{\mathsf{HBG}}$, pk and $x \in \{0,1\}^n$, run $(\widetilde{I}, \widetilde{r}_{\widetilde{I}}, \widetilde{\pi}_{\mathsf{HBM}}) \leftarrow \mathcal{S}_{\mathsf{HBM}}(1^\lambda, x)$ and $(\widetilde{\sigma}, \{\widetilde{\pi}_{\mathsf{HBG},i}\}_{i \in \widetilde{I}}) \leftarrow \mathcal{S}_{\mathsf{HBG},2}(\mathsf{st}_{\mathsf{HBG}}, \mathsf{pk}, \widetilde{I}, \widetilde{r}_{\widetilde{I}} \oplus \widetilde{s}_{\widetilde{I}})$. Output the simulated proof $\widetilde{\pi} = (\widetilde{\sigma}, \widetilde{I}, \widetilde{r}_{\widetilde{I}} \oplus \widetilde{s}_{\widetilde{I}}, \{\widetilde{\pi}_{\mathsf{HBG},i}\}_{i \in \widetilde{I}}, \widetilde{\pi}_{\mathsf{HBM}})$.

To complete the proof, we use the same hybrid structure as in the proof of Theorem 3.8:

- Hyb_0: This is the real distribution.
- Hyb_1: Same as Hyb_0, except the challenger uses $\mathcal{S}_{\mathsf{HBG},1}$ to generate the common reference string. It uses $\mathcal{S}_{\mathsf{HBG},2}$ to simulate the openings to the hidden-bits generator when responding to oracle queries.
- Hyb_2: Same as Hyb_1, except when responding to oracle queries, the challenger uses the simulator for the hidden-bits model NIZK to simulate the proofs. This is the simulated distribution.

By essentially the same argument as in the proof of Lemma 3.9, Hyb_0 and Hyb_1 are statistically indistinguishable if Π_{HBG} satisfies statistical simulation for malicious keys. Similarly, Hyb_1 and Hyb_2 are statistically indistinguishable if Π_{HBM} provides statistical zero-knowledge. This follows by an identical argument as in the proof of Lemma 3.10. □

B Publicly-Verifiable Statistical NIZK Arguments from HBG

In this section, we show that a (publicly-verifiable) hidden-bits generator that satisfies computational binding (Definition 4.10) yields a (publicly-verifiable NIZK) that satisfies non-adaptive computational soundness. The analysis is very similar to the proof of statistical soundness (Theorem 3.7). Here, we consider a variant of Construction 3.4 that is publicly-verifiable but has just a single mode. This is essentially the construction from [QRW19]. For completeness, we recall the specific construction:

Construction B.1 (NIZK from Publicly-Verifiable HBG)**.** Let $\mathcal{L} \subseteq \{0,1\}^n$ be an NP language with associated NP relation $\mathcal{R}$. We rely on the following building blocks:

- Let $\Pi_{\mathsf{HBG}} = (\mathsf{HBG.Setup}, \mathsf{HBG.GenBits}, \mathsf{HBG.Verify})$ be a publicly-verifiable hidden-bits generator with commitments of length $\ell = \ell(\lambda, \rho)$, where λ is the security parameter and ρ is the output length of the generator.
- Let $\Pi_{\mathsf{HBM}} = (\mathsf{HBM.Prove}, \mathsf{HBM.Verify})$ be a NIZK in the hidden-bits model for $\mathcal{L}$, and let $\rho = \rho(\lambda)$ be the length of the hidden-bits string for Π_{HBM}.

We construct a publicly-verifiable NIZK $\Pi_{\mathsf{NIZK}} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ for $\mathcal{L}$ as follows:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$: On input $\lambda \in \mathbb{N}$, sample $s \xleftarrow{R} \{0,1\}^\rho$. Run $\mathsf{crs}_{\mathsf{HBG}} \leftarrow \mathsf{HBG.Setup}(1^\lambda, 1^\rho, \mathsf{mode})$ and output $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$.
- $\mathsf{Prove}(\mathsf{crs}, x, w) \to \pi$: On input $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$, $x \in \{0,1\}^n$, and w, compute a hidden-bits string $(\sigma, r, \{\pi_{\mathsf{HBG},i}\}_{i \in \{1,\ldots,\rho\}}) \leftarrow \mathsf{HBG.GenBits}(\mathsf{crs}_{\mathsf{HBG}}, \mathsf{pk}_{\mathsf{HBG}})$, and an HBM proof $(I, \pi_{\mathsf{HBM}}) \leftarrow \mathsf{HBM.Prove}(1^\lambda, r \oplus s, x, w)$. Output $\pi = (\sigma, I, r_I, \{\pi_{\mathsf{HBG},i}\}_{i \in I}, \pi_{\mathsf{HBM}})$.
- $\mathsf{Verify}(\mathsf{crs}, x, \pi)$: On input $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$, $x \in \{0,1\}^n$, and $\pi = (\sigma, I, r_I, \{\pi_{\mathsf{HBG},i}\}_{i \in I}, \pi_{\mathsf{HBM}})$, output 1 if $\mathsf{HBM.Verify}(1^\lambda, I, r_I \oplus s_I, x, \pi_{\mathsf{HBM}}) = 1$ and $\mathsf{HBG.Verify}(\mathsf{crs}_{\mathsf{HBG}}, \sigma, i, r_i, \pi_{\mathsf{HBG},i}) = 1$ for all $i \in I$. Otherwise, output 0.

Theorem B.2 (Completeness)**.** *If Π_{HBM} is complete and Π_{HBG} is correct, then Π_{NIZK} from Construction B.1 is complete.*

Proof. Follows by a similar argument as the proof of Theorem 3.5. □

Theorem B.3 (Computational Soundness)**.** *If Π_{HBM} is statistically sound with soundness error $\varepsilon(\lambda)$, Π_{HBG} is is computationally binding (Definition 4.10), and $2^\ell \cdot \varepsilon = \mathsf{negl}(\lambda)$, then Π_{NIZK} from Construction B.1 satisfies non-adaptive computational soundness.*

Proof. The proof uses a similar hybrid structure as in the proof of Theorem 3.7, except we are in the non-adaptive setting. In particular, fix a statement $x \notin \mathcal{L}$, and let $\mathcal{E}$ be the extractor associated with Π_{HBG} (from Definition 4.10). Consider the following sequence of hybrid experiments:

- Hyb_0: This is the real soundness experiment, where the challenger begins by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ and gives $\mathsf{crs} = (\lambda, s, \mathsf{crs}_{\mathsf{HBG}})$ to $\mathcal{A}$. At the end of the experiment, $\mathcal{A}$ outputs a proof π^* and the output of the experiment is 1 if $\mathsf{Verify}(\mathsf{crs}, x, \pi^*) = 1$.
- Hyb_1: Same as Hyb_0, except the challenger samples $(\mathsf{st}_{\mathcal{E}}, \widetilde{\mathsf{crs}}_{\mathsf{HBG}}) \leftarrow \mathcal{E}_1(1^\lambda, 1^\rho)$ and uses $\widetilde{\mathsf{crs}} = (\lambda, s, \widetilde{\mathsf{crs}}_{\mathsf{HBG}})$ in place of crs.

- Hyb_2: Same as Hyb_1, except at the end of the experiment, after the adversary outputs $\pi^* = \left(\sigma^*, I^*, r^*_{I^*}, \{\pi^*_{\mathsf{HBG},i}\}_{i \in I^*}, \pi^*_{\mathsf{HBM}}\right)$, the challenger performs the following check:
  - Compute $r \leftarrow \mathcal{E}_2(\mathsf{st}_{\mathcal{E}}, \sigma^*)$. If $\mathsf{HBG.Verify}(\widetilde{\mathsf{crs}}_{\mathsf{HBG}}, \sigma^*, i, r^*_i, \pi^*_{\mathsf{HBG},i}) = 1$ for all $i \in I^*$ and $r_{I^*} \neq r^*_{I^*}$, then the challenger aborts the experiments and outputs $\bot$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of an execution of experiment Hyb_i with adversary $\mathcal{A}$. We now show that each adjacent pair of hybrids are computationally indistinguishable, and moreover, that $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.

Lemma B.4. *If Π_{HBG} satisfies computational binding, then for all efficient $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \overset{c}{\approx} \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. Follows immediately by the CRS indistinguishability property of $\mathcal{E}$. □

Lemma B.5. *If Π_{HBG} satisfies computational binding, then for all efficient $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \overset{c}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. The only difference between Hyb_1 and Hyb_2 is the additional check the challenger performs when computing $\mathsf{Verify}(\mathsf{crs}, x, \pi^*)$. But if $\mathcal{A}$ manages to output a commitment σ^* and an index $i \in [\rho]$ such that $\mathsf{HBG.Verify}(\widetilde{\mathsf{crs}}_{\mathsf{HBG}}, \sigma^*, i, r^*_i, \pi^*_{\mathsf{HBG},i}) = 1$ and $r_i \neq r^*_i$ where $r \overset{\mathrm{R}}{\leftarrow} \mathcal{E}_2(\mathsf{st}_{\mathcal{E}}, \sigma^*)$, then $(\sigma^*, i, r^*_i, \pi_{\mathsf{HBG},i})$ breaks the binding property of Π_{HBG}. Hence, if Π_{HBG} is computationally binding, then the probability that the additional check fails and the challenger aborts is negligible. □

Lemma B.6. *For all adversaries $\mathcal{A}$, $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.*

Proof. Follows by a similar argument as the proof of Lemma A.2. □

Thus, for all efficient adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \overset{c}{\approx} \mathsf{Hyb}_2(\mathcal{A})$ and $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$. Thus, the probability that $\mathsf{Hyb}_0(\mathcal{A})$ outputs 1 (meaning that $\mathcal{A}$ breaks non-adaptive soundness) is also negligible. □

Theorem B.7 (Statistical Zero-Knowledge)**.** *If Π_{HBM} satisfies statistical zero-knowledge and Π_{HBG} provides statistical simulation, then Π_{NIZK} from Construction B.1 is statistical zero-knowledge.*

Proof. Follows by a similar argument as the proof of Theorem 3.8. □

C Analysis of Constructions from k-Lin (Section 4)

In this section, we provide the analysis of the dual-mode hidden-bits generators from Section 4.

C.1 Analysis of Construction 4.3 (Dual-Mode HBG from k-Lin)

In this section, we give the proofs for the correctness and security theorems (Theorems 4.4 to 4.8) for the dual-mode hidden-bits generator from the k-Lin assumption (Construction 4.3).

Proof of Theorem 4.4 (Correctness). Fix $\lambda \in \mathbb{N}$, a polynomial $\rho = \rho(\lambda)$, an index $i \in [\rho]$, and a mode $\mathsf{mode} \in \{\mathsf{binding}, \mathsf{hiding}\}$. Let $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode})$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$, and $(\sigma, r, \{\pi_i\}_{i\in[\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs}, \mathsf{pk})$. By construction, $\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho})$, $\mathsf{pk} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_\rho})$, and $\mathsf{sk} = (a, \mathbf{b}_1, \ldots, \mathbf{b}_\rho)$. Moreover, $\mathbf{z}_i = \mathbf{w}_i a + \mathbf{V}\mathbf{b}_i$. In addition, $\sigma = g^{\mathbf{c}^\mathsf{T}} = g^{\mathbf{y}^\mathsf{T}\mathbf{V}}$, $r_i = H(g^{t_i})$, and $\pi_i = (g^{t_i}, g^{u_i})$, where $t_i = \mathbf{y}^\mathsf{T}\mathbf{w}_i$ and $u_i = \mathbf{y}^\mathsf{T}\mathbf{z}_i$ for some $\mathbf{y} \in \mathbb{Z}_p^{\rho+k}$. This means that

$$u_i = \mathbf{y}^\mathsf{T}\mathbf{z}_i = \mathbf{y}^\mathsf{T}(\mathbf{w}_i a + \mathbf{V}\mathbf{b}_i) = t_i a + \mathbf{c}^\mathsf{T}\mathbf{b}_i,$$

and the verification algorithm accepts. □

Proof of Theorem 4.5 (Succinctness). The length of a commitment in Construction 4.3 consists of $k = O(1)$ group elements, each of which can be represented by a string of length $\mathsf{poly}(\lambda)$. Thus, $|\sigma| = k \cdot \mathsf{poly}(\lambda) = \mathsf{poly}(\lambda)$. □

Proof of Theorem 4.6 (CRS Indistinguishability). Our analysis relies on the following corollary of the k-Lin assumption, which roughly says that an encoding of a random rank-k matrix is computationally indistinguishable for an encoding of a uniformly random matrix.

Lemma C.1 ([EHK+13])**.** *Suppose* GroupGen *is a prime-order group generator where k-Lin holds. Then for all polynomials $n = n(\lambda)$ where $k \le n$, the following two distributions are computationally indistinguishable:*

$$(\mathcal{G}, g^{\mathbf{V}}, g^{\mathbf{V}\mathbf{s}}) \overset{c}{\approx} (\mathcal{G}, g^{\mathbf{V}}, g^{\mathbf{w}}),$$

where $\mathcal{G} = (\mathbb{G}, p, g) \leftarrow \mathsf{GroupGen}(1^\lambda)$, $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{n\times k}$, $\mathbf{s} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ and $\mathbf{w} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^n$.

To complete the proof observe that the output of $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode})$ for $\mathsf{mode} \in \{\mathsf{binding}, \mathsf{hiding}\}$ differ only in how the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_\rho$ are sampled. In both cases, Setup starts by sampling $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(\rho+k)\times k}$. Then, if $\mathsf{mode} = \mathsf{binding}$, $\mathbf{w}_i \leftarrow \mathbf{V}\mathbf{s}_i$ for $\mathbf{s}_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$, while if $\mathsf{mode} = \mathsf{hiding}$, $\mathbf{w}_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\rho+k}$. By Lemma C.1, these two distributions are computationally indistinguishable, and the claim now follows by a standard hybrid argument. □

Proof of Theorem 4.7 (Statistical Binding). We first define the (inefficient) Open algorithm as follows:

- $\mathsf{Open}(\mathsf{crs}, \sigma) \to r$: On input a $\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho})$ and a commitment $\sigma = g^{\mathbf{c}^\mathsf{T}}$, the open algorithm first recovers $\mathbf{V}, \mathbf{w}_1, \ldots, \mathbf{w}_\rho$ by solving the discrete logarithm problem over $\mathbb{G}$. For each $i \in [\rho]$, it checks that $\mathbf{w}_i = \mathbf{V}\mathbf{s}_i$ for some $\mathbf{s}_i \in \mathbb{Z}_p^k$, and outputs $\perp$ if not. If all checks pass, it computes $r_i \leftarrow H(g^{\mathbf{c}^\mathsf{T}\mathbf{s}_i})$ for each $i \in [\rho]$ and outputs r.

To complete the proof, we use a hybrid argument:

- Hyb_0: This is the real binding experiment. Namely, the challenger starts by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{binding})$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$, and gives $(\mathsf{crs}, \mathsf{pk})$ to the adversary. By construction,

$$\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{V}\mathbf{s}_1}, \ldots, g^{\mathbf{V}\mathbf{s}_\rho}) \quad \text{and} \quad \mathsf{pk} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_\rho}),$$

where $\mathbf{z}_i = \mathbf{V}(\mathbf{s}_i a + \mathbf{b}_i)$. The adversary can then make queries to the verification oracle. On each query (σ, i, r_i, π_i), the challenger replies with $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$. At the end of the game, the adversary outputs a tuple $(\sigma^*, i^*, r^*, \pi^*)$. The output of the experiment is 1 if $r^* \neq r_i$ where $r \leftarrow \mathsf{Open}(\mathsf{crs}, \sigma^*)$ and $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma^*, i^*, r^*, \pi^*) = 1$.

- Hyb_1: Same as Hyb_0, except the challenger samples $\mathbf{x}_1, \ldots, \mathbf{x}_\rho \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ and computes $\mathbf{z}_i \leftarrow \mathbf{V}\mathbf{x}_i$. It then samples $a \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ and sets $\mathbf{b}_i \leftarrow \mathbf{x}_i - \mathbf{s}_i a$.
- Hyb_2: Same as Hyb_1, except the challenger implements $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$ by computing $\mathsf{Verify}^*(\sigma, r_i, \pi_i, \mathbf{s}_i, \mathbf{x}_i)$, where on input $\sigma = g^{\mathbf{c}^\mathsf{T}}$, $r_i \in \{0,1\}$, $\pi_i = (g^{t_i}, g^{u_i})$, $\mathbf{s}_i \in \mathbb{Z}_p^k$, $\mathbf{x}_i \in \mathbb{Z}_p^k$, $\mathsf{Verify}^*(\sigma, \pi_i, r_i, \mathbf{s}_i, \mathbf{x}_i)$ outputs 1 if and only if

$$r_i = H(g^{t_i}) \quad \text{and} \quad t_i = \mathbf{c}^\mathsf{T}\mathbf{s}_i \quad \text{and} \quad u_i = \mathbf{c}^\mathsf{T}\mathbf{x}_i.$$

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output of Hyb_i with adversary $\mathcal{A}$. We show that the output distributions of each adjacent pair of hybrid experiments are statistically indistinguishable, and moreover, $\mathsf{Hyb}_2(\mathcal{A})$ output 0 with overwhelming probability for all adversaries $\mathcal{A}$.

Lemma C.2. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \equiv \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. Follows by construction. □

Lemma C.3. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. We use a hybrid argument. The challenger evaluates Verify at most $Q+1$ times, where $Q = \mathsf{poly}(\lambda)$ is the bound on the number of queries the adversary makes (the challenger evaluates Verify also at the end to determine the output of the experiment). We define a sequence of intermediate hybrids $\mathsf{Hyb}_{1,j}$ for $j \in \{0, \ldots Q+1\}$, where in hybrid $\mathsf{Hyb}_{1,j}$, the first j queries are handled according to the specification in Hyb_2 while the remaining queries are handled according to the specification in Hyb_1. By construction, $\mathsf{Hyb}_1 \equiv \mathsf{Hyb}_{1,0}$ and $\mathsf{Hyb}_2 \equiv \mathsf{Hyb}_{1,Q+1}$. Consider $\mathsf{Hyb}_{1,j-1}$ and $\mathsf{Hyb}_{1,j}$ for $j \in [Q+1]$. These two experiments only differ in how the challenger computes the output for the j^{th} Verify call. Let $(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$ be the arguments to the j^{th} Verify call, where $\sigma = g^{\mathbf{c}^T}$ and $\pi_i = (g^{t_i}, g^{u_i})$. Consider the behavior in $\mathsf{Hyb}_{1,j-1}$. Here, the verifier accepts only if

$$r_i = H(g^{t_i}) \quad \text{and} \quad u_i = t_i a + \mathbf{c}^\mathsf{T}\mathbf{b}_i = \mathbf{c}^\mathsf{T}(\mathbf{s}_i a + \mathbf{b}_i) + (t_i - \mathbf{c}^\mathsf{T}\mathbf{s}_i)a = \mathbf{c}^\mathsf{T}\mathbf{x}_i + (t_i - \mathbf{c}^\mathsf{T}\mathbf{s}_i)a.$$

We consider two possibilities:

- If $t_i = \mathbf{c}^\mathsf{T}\mathbf{s}_i$, then Verify in $\mathsf{Hyb}_{1,j-1}$ outputs 1 if and only if $r_i = H(g^{t_i})$ and $u_i = \mathbf{c}^\mathsf{T}\mathbf{x}_i$. This is identical to the behavior of Verify^* in $\mathsf{Hyb}_{1,j}$.
- If $t_i \neq \mathbf{c}^\mathsf{T}\mathbf{s}_i$, then the output in $\mathsf{Hyb}_{1,j-1}$ is 1 only if $a = (u_i - \mathbf{c}^\mathsf{T}\mathbf{x}_i)(t_i - \mathbf{c}^\mathsf{T}\mathbf{s}_i)^{-1}$. In $\mathsf{Hyb}_{1,j-1}$, neither the public parameters (crs and pk) nor the responses to the first $j-1$ queries depend on the value of a (in particular, Verify^* is independent of a, and thus, leaks no information about a). Thus, the challenger can sample $a \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ *after* the adversary has chosen $\mathbf{c}$, t_i, and u_i. In this case, Verify outputs 1 with probability $1/p = \mathsf{negl}(\lambda)$. Thus, with overwhelming probability, the output in $\mathsf{Hyb}_{1,j-1}$ is 0. This is the output of Verify^* in $\mathsf{Hyb}_{1,j}$.

Since the outputs of Verify in Hyb_1 and Verify^* in Hyb_2 are statistically indistinguishable for each query, the claim now follows by a hybrid argument. To conclude the proof, it suffices to show that the output in Hyb_2 is always 0. Let $(\sigma^*, i^*, r^*, \pi^*)$ be the adversary's output in Hyb_2. The output in Hyb_2 is 1 only if $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma^*, i^*, r^*, \pi^*) = 1$ and $r_{i^*} \neq r^*$ where $r \leftarrow \mathsf{Open}(\mathsf{crs}, \sigma^*)$. Write $\sigma^* = g^{(\mathbf{c}^*)^\mathsf{T}}$ and $\pi^* = (g^{t_i^*}, g^{u_i^*})$. By definition of Open, $r_{i^*} = H(g^{(\mathbf{c}^*)^\mathsf{T}\mathbf{s}_i})$. In Hyb_3, the Verify function outputs 1 only if $t_i^* = (\mathbf{c}^*)^\mathsf{T}\mathbf{s}_i$ and

$$r^* = H(g^{t_i^*}) = H(g^{(\mathbf{c}^*)^\mathsf{T}\mathbf{s}_i}) = r_{i^*}.$$

But then, the output of Hyb_2 is 0. Since the output of Hyb_0 and Hyb_2 are statistically indistinguishable, this means that the output in Hyb_0 is also 0 with overwhelming probability. □

Theorem 4.7 now follows by a hybrid argument. □

Proof of Theorem 4.8 (Statistical Simulation). We construct a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ as follows:

- $\mathcal{S}_1(1^\lambda, 1^\rho) \to (\mathsf{st}_\mathcal{S}, \widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}})$: Sample $\mathcal{G} = (\mathbb{G}, p, g) \leftarrow \mathsf{GroupGen}(1^\lambda)$, $H \xleftarrow{\mathrm{R}} \mathcal{H}$, $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(\rho+k)\times k}$, $\mathbf{w}_1, \ldots, \mathbf{w}_\rho \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\rho+k}$, $a \xleftarrow{\mathrm{R}} \mathbb{Z}_p$, and $\mathbf{b}_1, \ldots, \mathbf{b}_\rho \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$. For each $i \in [\rho]$, let $\mathbf{z}_i \leftarrow \mathbf{w}_i a + \mathbf{V}\mathbf{b}_i$. In the following, we will write

  $$\mathbf{W} = [\ \mathbf{w}_1 \mid \cdots \mid \mathbf{w}_\rho \mid \mathbf{V}\ ] \in \mathbb{Z}_p^{(\rho+k)\times(\rho+k)}. \tag{C.1}$$

  Output $\widetilde{\mathsf{crs}} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho})$, $\widetilde{\mathsf{pk}} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_\rho})$, $\widetilde{\mathsf{sk}} = (a, \mathbf{b}_1, \ldots, \mathbf{b}_\rho)$, and $\mathsf{st}_\mathcal{S} = (\widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \mathbf{W})$

- $\mathcal{S}_2(\mathsf{st}_\mathcal{S}, I, r_I) \to (\widetilde{\sigma}, \{\widetilde{\pi}_i\}_{i\in I})$: On input $\mathsf{st}_\mathcal{S} = (\widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \mathbf{W})$ where

  $$\widetilde{\mathsf{crs}} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho}) \quad \text{and} \quad \widetilde{\mathsf{pk}} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_\rho}),$$

  a set of indices $I \subseteq [\rho]$, and a bitstring $r_I \in \{0,1\}^{|I|}$, the simulator samples a vector $\mathbf{t} \in \mathbb{Z}_p^\rho$ as follows:

  - For each $i \in I$, sample $t_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ conditioned on $H(g^{t_i}) = r_i$. Specifically, the simulator repeatedly samples $t_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ until finding one that satisfies $H(g^{t_i}) = r_i$. If no such t_i is found after λ iterations, then the simulator aborts and outputs $\perp$.
  - For the remaining indices $i \in [\rho] \setminus I$, sample $t_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$.

  Next, it samples $\mathbf{c} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ and sets $\mathbf{y}^\mathsf{T} \leftarrow [\ \mathbf{t}^\mathsf{T} \mid \mathbf{c}^\mathsf{T}\ ] \cdot \mathbf{W}^{-1}$ (outputting $\perp$ if $\mathbf{W}$ is not invertible). It computes $\widetilde{\sigma} = g^{\mathbf{c}^\mathsf{T}}$ and $g^{u_i} \leftarrow g^{\mathbf{y}^\mathsf{T}\mathbf{z}_i}$ for each $i \in I$. It output $\widetilde{\sigma}$ and $\{\widetilde{\pi}_i\}_{i\in I}$ where $\widetilde{\pi}_i = (g^{t_i}, g^{u_i})$.

To show that $\mathsf{ExptHide}[\mathcal{A}, 0]$ and $\mathsf{ExptHide}[\mathcal{A}, 1]$ are statistically indistinguishable, we use a hybrid argument:

- Hyb_0: This is the distribution in $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 0]$. Namely, the challenger begins by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$. For each challenge query, the challenger first samples $(\sigma, r, \{\pi_i\}_{i\in[\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs}, \mathsf{pk})$ and gives r to $\mathcal{A}$ to receive a set $I \subseteq [\rho]$. It then replies with σ and $\{\pi_i\}_{i\in I}$.

- Hyb_1: Same as Hyb_0, except the challenger computes $(\mathsf{st}_{\mathcal{S}}, \widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}}) \leftarrow \mathcal{S}_1(1^\lambda, 1^\rho)$ and uses $\widetilde{\mathsf{crs}}$, $\widetilde{\mathsf{pk}}$, and $\widetilde{\mathsf{sk}}$ in place of crs, pk, and sk, respectively.

  Specifically, the challenger samples $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(\rho+k)\times k}$, $\mathbf{w}_1, \ldots, \mathbf{w}_\rho \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\rho+k}$, $a \xleftarrow{\mathrm{R}} \mathbb{Z}_p$, and $\mathbf{b}_1, \ldots, \mathbf{b}_\rho \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$. For each $i \in [\rho]$, it sets $\mathbf{z}_i \leftarrow \mathbf{w}_i a + \mathbf{V}\mathbf{b}_i$. It sets $\widetilde{\mathsf{crs}} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho})$, $\widetilde{\mathsf{pk}} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_\rho})$, and $\widetilde{\mathsf{sk}} = (a, \mathbf{b}_1, \ldots, \mathbf{b}_\rho)$. On each challenge query, the challenger samples $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\rho+k}$ and computes $[\ \mathbf{t}^\mathsf{T} \mid \mathbf{c}^\mathsf{T}\ ] \leftarrow \mathbf{y}^\mathsf{T}\mathbf{W}$, $r_i \leftarrow H(g^{t_i})$, and $g^{u_i} \leftarrow g^{\mathbf{y}^\mathsf{T}\mathbf{z}_i}$ for each $i \in [\rho]$, where $\mathbf{W}$ is defined in Eq. (C.1). The challenger sends r to the adversary and receives a set $I \subseteq [t]$. The challenger replies with $\sigma = g^{\mathbf{c}^\mathsf{T}}$ and $\{\pi_i\}_{i\in I} = \{(g^{t_i}, g^{u_i})\}_{i\in I}$.

- Hyb_2: Same as Hyb_1, except when responding to the challenge queries, instead of sampling $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\rho+k}$, the challenger samples $\mathbf{t} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^\rho$ and $\mathbf{c} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$. It sets $\mathbf{y}^\mathsf{T} \leftarrow [\ \mathbf{t}^\mathsf{T} \mid \mathbf{c}^\mathsf{T}\ ] \cdot \mathbf{W}^{-1}$ (and outputs $\perp$ if $\mathbf{W}$ is not invertible). All remaining components are constructed as in Hyb_1.

- Hyb_3: Same as Hyb_2 except when responding to the challenge queries, the challenger first samples $r \xleftarrow{\mathrm{R}} \{0,1\}^k$. For each $i \in [\rho]$, it samples $t_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ conditioned on $H(g^{t_i}) = r_i$. The challenger uses the same rejection procedure for this step as in $\mathcal{S}_2$ to implement this. It samples $\mathbf{c} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ and the remaining components as in Hyb_2.

- Hyb_4: Same as Hyb_3 except when responding to the challenge queries, the challenger samples $\mathbf{t}$ *after* it receives the challenge set. In particular, on each query, after the challenger receives the set $I \subseteq [\rho]$, it samples $t_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ conditioned on $H(g^{t_i}) = r_i$ if $i \in I$ and $t_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ otherwise. All remaining components are constructed as in Hyb_3. This is the distribution in $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 1]$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of experiment Hyb_i with adversary $\mathcal{A}$. We now show that for all adversaries $\mathcal{A}$, the output distributions of each consecutive pair of hybrids are either statistically indistinguishable or identically distributed.

Lemma C.4. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \equiv \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. Since $\mathcal{S}_1(1^\lambda, 1^\rho)$ samples $\widetilde{\mathsf{crs}}$, $\widetilde{\mathsf{pk}}$, and $\widetilde{\mathsf{sk}}$ using the same procedure as Setup and KeyGen, the output distributions of hybrids Hyb_0 and Hyb_1 are identically distributed. □

Lemma C.5. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. With overwhelming probability, $\mathbf{W}$ is full-rank and thus invertible. In this case, the distributions of $\mathbf{y}$ in Hyb_1 and Hyb_2 are identical. Thus, the output distributions of Hyb_1 and Hyb_2 are statistically indistinguishable. □

Lemma C.6. *If $\mathcal{H}$ is statistically uniform, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_2(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_3(\mathcal{A})$.*

Proof. Since H is sampled from a hash family that satisfies statistical uniformity, the distribution of $H(g^{t_i})$ when $t_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ is statistically close to uniform over $\{0,1\}$. Thus, in Hyb_2, each bit r_i is statistically close to uniform. It suffices to argue that the sampling algorithm in Hyb_3 does not abort. In this is the case, then the two distributions are statistically close. Again, using the fact that $\mathcal{H}$ is statistically uniform, for a random $t_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$, $\Pr[H(g^{t_i}) = r_i] \geq 1/2 - \mathsf{negl}(\lambda)$ for any $r_i \in \{0,1\}$. Thus, with overwhelming probability, the challenger successfully finds a t_i after λ independent attempts. Since $\rho = \mathsf{poly}(\lambda)$ and the adversary makes at most $q = \mathsf{poly}(\lambda)$ queries, the claim now follows by a union bound. □

Lemma C.7. *For all adversaries* $\mathcal{A}$, $\mathsf{Hyb}_3(\mathcal{A}) \equiv \mathsf{Hyb}_4(\mathcal{A})$.

Proof. First, the challenger's first message to the adversary in both experiments is $(\widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}})$ where $\widetilde{\mathsf{crs}} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_\rho})$, $\widetilde{\mathsf{pk}} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_\rho})$, and $\widetilde{\mathsf{sk}} = (a, \mathbf{b}_1, \ldots, \mathbf{b}_\rho)$. These components are identically distributed in the two experiments. We consider the challenge queries. On each challenge query, in both Hyb_3 and Hyb_4, the challenger sends the adversary a random string $r \xleftarrow{R} \{0,1\}^k$. To conclude the proof, we argue that the challenger's response to each challenge query is identically distributed in the two experiments. In particular, after the adversary outputs a set $I \subseteq [\rho]$, the challenger replies with a commitment $\sigma = g^{\mathbf{c}^\mathsf{T}}$ and a collection of proofs $\{\pi_i\}_{i \in I} = \{(g^{t_i}, g^{u_i})\}_{i \in I}$. We show that $\mathbf{c}, t_i, u_i$ for $i \in I$ in Hyb_3 and Hyb_4 are identically distributed:

- In Hyb_3 and Hyb_4, for all $i \in I$, the challenger samples $t_i \xleftarrow{R} \mathbb{Z}_p$ subject to $H(g^{t_i}) = r_i$. Thus, the distribution of t_i for each $i \in I$ is identically distributed in the two experiments.
- In Hyb_3 and Hyb_4, $\mathbf{c} \in \mathbb{Z}_p^k$ is independent and uniformly distributed.
- In Hyb_3 and Hyb_4, $u_i = \mathbf{y}^\mathsf{T}\mathbf{z}_i = \mathbf{y}^\mathsf{T}(\mathbf{w}_i a + \mathbf{V}\mathbf{b}_i) = t_i a + \mathbf{c}^\mathsf{T}\mathbf{b}_i$. Since $a, \mathbf{b}_i, t_i, \mathbf{c}$ are all identically distributed in the two experiments for $i \in I$, we conclude that u_i is identically distributed in Hyb_3 and Hyb_4 for all $i \in I$. □

Theorem 4.8 now follows by a hybrid argument. □

Remark C.8 (Perfect Simulation in Hiding Mode)**.** A modified version of Construction 4.3 can be shown to satisfy *perfect* simulation in hiding mode. Specifically, we make the following adjustments:

- Modify $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$ so that it always outputs $\mathbf{V}, \mathbf{w}_1, \ldots, \mathbf{w}_\rho$ such that the matrix $\mathbf{W}$ in Eq. (C.1) is full-rank. This property already holds with overwhelming probability, so this only introduces a negligible loss to the CRS indistinguishability property. With this change, Hyb_1 and Hyb_2 in the proof of Theorem 4.8 are identically distributed.
- Replace the hash family $\mathcal{H}$ with one that is *perfectly* uniform and one where there is a procedure to *exactly* sample $t_i \xleftarrow{R} \mathbb{Z}_p$ such that $H(g^{t_i}) = r_i$.

  Currently, $\mathcal{H}$ is a hash family from a *prime-order* group $\mathbb{G}$ to $\{0,1\}$, so perfect uniformity is impossible. However, we can modify the construction to use a hash function with domain $\mathbb{G} \setminus \{g^0\}$, where $g^0 \in \mathbb{G}$ denotes the identity element. To enforce this, we modify the $\mathsf{GenBits}$ algorithm to sample $\mathbf{y} \xleftarrow{R} \mathbb{Z}_p^{\rho+k}$ subject to the restriction that $t_i = (\mathbf{y}^\mathsf{T}\mathbf{W})_i \neq 0$ for all $i \in [\rho]$. A randomly-sampled $\mathbf{y}$, will satisfy this property with overwhelming probability (since $1/p = \mathsf{negl}(\lambda)$), and thus, this can only change the distribution of $\mathsf{GenBits}$ by a negligible amount. With this modification and assuming that $\mathcal{H}$ is a perfectly uniform family of hash functions from $\mathbb{G} \setminus \{[0]\}$ to $\{0,1\}$, and that there is an efficient algorithm to sample $t_i \xleftarrow{R} \mathbb{Z}_p$ such that $H(g^{t_i}) = r_i$, then hybrids Hyb_2 and Hyb_3 in the proof of Theorem 4.8 are identically distributed. As we show in Section 6, it is straightforward to construct a hash function with these properties when $\mathbb{G}$ is an elliptic curve group over a finite field (see Remark 6.3).

With these modifications, each pair of adjacent hybrids in the proof of Theorem 4.8 is identically distributed, and we conclude that Π_{HBG} satisfies perfect simulation in hiding mode. Combined with Construction 3.4, this gives a dual-mode NIZK with *perfect* zero-knowledge in hiding mode.

C.2 Analysis of Construction 4.13 (Publicly-Verifiable HBG from Pairings)

In this section, we give the proofs for the correctness and security theorems (Theorems 4.14 to 4.17) for the publicly-verifiable hidden-bits generator from pairings (Construction 4.13).

Proof of Theorem 4.14 (Correctness). Fix $\lambda \in \mathbb{N}$, a polynomial $\rho = \rho(\lambda)$, and an index $i \in [\rho]$. Let $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho)$ and $(\sigma, r, \{\pi_i\}_{i\in[\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs})$. Consider the value of $\mathsf{Verify}(\mathsf{crs}, \sigma, i, r_i, \pi_i)$. First, by construction,

$$\mathsf{crs} = \left(\mathcal{G}, H, g_1^{\mathbf{V}}, g_2^{\hat{\mathbf{a}}^\mathsf{T}}, g_2^{\mathbf{D}}, \left\{g_1^{\mathbf{w}_i}, g_1^{\mathbf{Z}_i}, g_2^{\hat{\mathbf{B}}_i}\right\}_{i\in[\rho]}\right),$$

$\hat{\mathbf{a}}^\mathsf{T} = \mathbf{a}^\mathsf{T}\mathbf{D}$, $\mathbf{Z}_i = \mathbf{w}_i\mathbf{a}^\mathsf{T} + \mathbf{V}\mathbf{B}_i$, and $\hat{\mathbf{B}}_i = \mathbf{B}_i^\mathsf{T}\mathbf{D}$. In addition, $\sigma = g_1^{\mathbf{c}^\mathsf{T}} = g_1^{\mathbf{y}^\mathsf{T}\mathbf{V}}$, $r_i = H(g_1^{t_i})$, and $\pi_i = (g_1^{t_i}, g_1^{\mathbf{u}_i^\mathsf{T}})$, where

$$g_1^{t_i} = g_1^{\mathbf{y}^\mathsf{T}\mathbf{w}_i} \quad \text{and} \quad g_1^{\mathbf{u}_i^\mathsf{T}} = g_1^{\mathbf{y}^\mathsf{T}\mathbf{Z}_i} = g_1^{\mathbf{y}^\mathsf{T}(\mathbf{w}_i\mathbf{a}^\mathsf{T}+\mathbf{V}\mathbf{B}_i)}.$$

It suffices to check Eq. (4.3) holds. By construction,

$$e(g_1^{t_i}, g_2^{\hat{\mathbf{a}}^\mathsf{T}}) \cdot e(g_1^{\mathbf{c}^\mathsf{T}}, g_2^{\hat{\mathbf{B}}_i}) = e(g_1, g_2)^{(\mathbf{y}^\mathsf{T}\mathbf{w}_i)(\mathbf{a}^\mathsf{T}\mathbf{D})+(\mathbf{y}^\mathsf{T}\mathbf{V})(\mathbf{B}_i\mathbf{D})} = e(g_1, g_2)^{\mathbf{y}^\mathsf{T}(\mathbf{w}_i\mathbf{a}^\mathsf{T}+\mathbf{V}\mathbf{B}_i)\mathbf{D}} = e(g_1^{\mathbf{u}_i^\mathsf{T}}, g_2^{\mathbf{D}}),$$

and Verify accepts. □

Proof of Theorem 4.15 (Succinctness). The length of a commitment in Construction 4.13 consists of $k = O(1)$ elements in $\mathbb{G}_1$, each of which can be represented by a string of length $\mathsf{poly}(\lambda)$. Thus, $|\sigma| = k \cdot \mathsf{poly}(\lambda) = \mathsf{poly}(\lambda)$. □

Proof of Theorem 4.16 (Computational Binding). We begin by constructing an efficient extractor $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ as follows:

- $\mathcal{E}_1(1^\lambda, 1^\rho) \to (\mathsf{st}_\mathcal{E}, \widetilde{\mathsf{crs}})$: Sample $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathsf{PairingGroupGen}(1^\lambda)$ and $H \xleftarrow{\mathrm{R}} \mathcal{H}$. Sample $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(\rho+k)\times k}$. For all $i \in [\rho]$, sample $\mathbf{s}_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ and set $\mathbf{w}_i \leftarrow \mathbf{V}\mathbf{s}_i$. Sample verification components $\mathbf{a} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k+1}$, $\mathbf{B}_1, \ldots, \mathbf{B}_\rho \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k\times(k+1)}$, $\mathbf{d} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$, and set $\mathbf{D}$ as in Eq. (4.2). Then, set $\hat{\mathbf{a}}^\mathsf{T} \leftarrow \mathbf{a}^\mathsf{T}\mathbf{D}$ and for each $i \in [\rho]$, let $\mathbf{Z}_i \leftarrow \mathbf{w}_i\mathbf{a}^\mathsf{T} + \mathbf{V}\mathbf{B}_i$ and $\hat{\mathbf{B}}_i \leftarrow \mathbf{B}_i\mathbf{D}$ as in Setup. Output $\widetilde{\mathsf{crs}} = \left(\mathcal{G}, H, g_1^{\mathbf{V}}, g_2^{\hat{\mathbf{a}}^\mathsf{T}}, g_2^{\mathbf{D}}, \left\{g_1^{\mathbf{w}_i}, g_1^{\mathbf{Z}_i}, g_2^{\hat{\mathbf{B}}_i}\right\}_{i\in[\rho]}\right)$ and $\mathsf{st}_\mathcal{E} = (H, \mathbf{s}_1, \ldots, \mathbf{s}_k)$.
- $\mathcal{E}_2(\mathsf{st}_\mathcal{E}, \sigma) \to r$: On input $\mathsf{st}_\mathcal{E} = (H, \mathbf{s}_1, \ldots, \mathbf{s}_k)$ and a commitment $\sigma = g_1^{\mathbf{c}^\mathsf{T}}$, compute $r_i \leftarrow H(g_1^{\mathbf{c}^\mathsf{T}\mathbf{s}_i})$ for each $i \in [\rho]$. Output r. This is a deterministic algorithm by construction.

We now show that $\mathcal{E}$ satisfies the two required properties.

CRS indistinguishability. The only difference in the CRS output by Setup and that output by $\mathcal{E}_1$ is in how the $\mathbf{w}_i$ are sampled (from $\mathbf{w}_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ in Setup to $\mathbf{w}_i \leftarrow \mathbf{V}\mathbf{s}_i$ where $\mathbf{s}_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ in $\mathcal{E}_1$). This is precisely the distinction between the CRS in the hiding mode and the CRS in the binding mode in Construction 4.3. As such, the proof follows via the same argument as in the proof of Theorem 4.6 (by appealing to k-Lin in $\mathbb{G}_1$).

Binding. We use a hybrid argument:

- Hyb_0: This is the real binding experiment. Namely, the challenger starts by sampling $(\mathsf{st}_{\mathcal{E}}, \widetilde{\mathsf{crs}}) \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho)$ and gives $\widetilde{\mathsf{crs}} = \big(\mathcal{G}, H, g_1^{\mathbf{V}}, g_2^{\hat{\mathbf{a}}^\mathsf{T}}, g_2^{\mathbf{D}}, \{g_1^{\mathbf{w}_i}, g_1^{\mathbf{Z}_i}, g_2^{\hat{\mathbf{B}}_i}\}_{i\in[\rho]}\big)$ to the adversary. Here, $\mathbf{w}_i = \mathbf{V}\mathbf{s}_i$, $\hat{\mathbf{a}}^\mathsf{T} = \mathbf{a}^\mathsf{T}\mathbf{D}$, $\mathbf{Z}_i = \mathbf{w}_i\mathbf{a}^\mathsf{T} + \mathbf{V}\mathbf{B}_i$, and $\hat{\mathbf{B}}_i = \mathbf{B}_i\mathbf{D}$. The adversary then outputs a tuple $(\sigma^*, i^*, r^*, \pi^*)$. The output of the experiment is 1 if $r^* \neq r_i$ where $r \leftarrow \mathcal{E}_2(\mathsf{st}_{\mathcal{E}}, \sigma^*)$ and $\mathsf{Verify}(\widetilde{\mathsf{crs}}, \sigma^*, i^*, r^*, \pi^*) = 1$. Otherwise, the output is 0.
- Hyb_1: Same as Hyb_0 except after the adversary outputs its tuple $(\sigma^*, i^*, r^*, \pi^*)$, the challenger performs the following additional check. First, write $\sigma^* = g_1^{(\mathbf{c}^*)^\mathsf{T}}$ and $\pi^* = (g_1^{t^*}, g_1^{(\mathbf{u}^*)^\mathsf{T}})$. The output of Hyb_1 is 0 if $\mathsf{Verify}(\widetilde{\mathsf{crs}}, \sigma^*, i^*, r^*, \pi^*) = 1$ and $t^*\mathbf{a}^\mathsf{T} + (\mathbf{c}^*)^\mathsf{T}\mathbf{B}_{i^*} \neq (\mathbf{u}^*)^\mathsf{T}$. Otherwise, the output of Hyb_1 is computed as in Hyb_0.

We write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output of experiment Hyb_i with adversary $\mathcal{A}$. To complete the proof, we show that for all efficient adversaries $\mathcal{A}$, the output distributions of $\mathsf{Hyb}_0(\mathcal{A})$ and $\mathsf{Hyb}_1(\mathcal{A})$ are computationally indistinguishable, and moreover, that $\mathsf{Hyb}_1(\mathcal{A})$ outputs 1 with negligible probability. This means that the output in Hyb_0 (the real binding experiment) is 1 with negligible probability, which proves the claim.

Lemma C.9. *Suppose the k-KerLin assumption holds in $\mathbb{G}_2$ with respect to* PairingGroupGen*. Then, for all efficient adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \overset{c}{\approx} \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. The only difference between $\mathsf{Hyb}_0(\mathcal{A})$ and $\mathsf{Hyb}_1(\mathcal{A})$ is the additional check the challenger performs at the end of the experiment. Suppose there exists an efficient adversary $\mathcal{A}$ such that $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] > \Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1] + \varepsilon$ for some non-negligible ε. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that breaks the k-KerLin assumption:

1. Algorithm $\mathcal{B}$ receives the k-KerLin challenge $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ and a matrix $g_2^{\mathbf{D}}$ from the challenger (where $\mathbf{D} \in \mathbb{Z}_p^{(k+1)\times k}$ has the form in Eq. (4.2)). It uses the challenge to simulate a CRS as follows. First, it samples $H \xleftarrow{\mathrm{R}} \mathcal{H}$ and $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(\rho+k)\times k}$, and vectors $\mathbf{s}_1, \ldots, \mathbf{s}_k \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$. It also samples verification components $\mathbf{a} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k+1}$, $\mathbf{B}_1, \ldots, \mathbf{B}_\rho \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k\times(k+1)}$. Finally, it sets
$$\widetilde{\mathsf{crs}} = \big(\mathcal{G}, H, g_1^{\mathbf{V}}, g_2^{\mathbf{a}^\mathsf{T}\mathbf{D}}, g_2^{\mathbf{D}}, \{g_1^{\mathbf{V}\mathbf{s}_i}, g_1^{\mathbf{V}\mathbf{s}_i\mathbf{a}^\mathsf{T}+\mathbf{V}\mathbf{B}_i}, g_2^{\mathbf{B}_i\mathbf{D}}\}_{i\in[\rho]}\big).$$
Note that $\mathcal{B}$ can efficiently compute the components $g_2^{\mathbf{a}^\mathsf{T}\mathbf{D}}$ and $g_2^{\mathbf{B}_i\mathbf{D}}$ from the challenge component $g_2^{\mathbf{D}}$ since it knows $\mathbf{a}$ and $\mathbf{B}_i$. The challenger gives $\widetilde{\mathsf{crs}}$ to $\mathcal{A}$.
2. Algorithm $\mathcal{A}$ outputs $\sigma^* = g_1^{(\mathbf{c}^*)^\mathsf{T}}$, $i^* \in [\rho]$, $r^* \in \{0,1\}$, and $\pi^* = \big(g_1^{t^*}, g_1^{(\mathbf{u}^*)^\mathsf{T}}\big)$.
3. Algorithm $\mathcal{B}$ outputs the group element $g_1^{t^*\mathbf{a}^\mathsf{T}+(\mathbf{c}^*)^\mathsf{T}\mathbf{B}_{i^*}-(\mathbf{u}^*)^\mathsf{T}}$. This can be computed efficiently given $g_1^{t^*}$, $g_1^{(\mathbf{c}^*)^\mathsf{T}}$, $g_1^{(\mathbf{u}^*)^\mathsf{T}}$, $\mathbf{a}$, and $\mathbf{B}_{i^*}$, all of which are known to $\mathcal{B}$.

By construction, $\mathcal{B}$ perfectly simulates $\widetilde{\mathsf{crs}}$ according to the specification in Hyb_0. Thus, with probability ε, algorithm $\mathcal{A}$ will output $g_1^{(\mathbf{c}^*)^\mathsf{T}}$, $g_1^{t^*}$, and $g_1^{(\mathbf{u}^*)^\mathsf{T}}$ such that $\mathsf{Verify}(\widetilde{\mathsf{crs}}, \sigma^*, i^*, r^*, \pi^*) = 1$ and $t^*\mathbf{a}^\mathsf{T} + (\mathbf{c}^*)^\mathsf{T}\mathbf{B}_{i^*} \neq (\mathbf{u}^*)^\mathsf{T}$. If the output of Verify is 1, then it must be the case that
$$e\big(g_1^{t^*}, g_2^{\mathbf{a}^\mathsf{T}\mathbf{D}}\big) \cdot e\big(g_1^{(\mathbf{c}^*)^\mathsf{T}}, g_2^{\mathbf{B}_{i^*}\mathbf{D}}\big) = e\big(g_1^{(\mathbf{u}^*)^\mathsf{T}}, g_2^{\mathbf{D}}\big),$$

or equivalently,

$$t^*\mathbf{a}^\mathsf{T}\mathbf{D} + (\mathbf{c}^*)^\mathsf{T}\mathbf{B}_{i^*}\mathbf{D} = (t^*\mathbf{a}^\mathsf{T} + (\mathbf{c}^*)^\mathsf{T}\mathbf{B}_{i^*})\mathbf{D} = (\mathbf{u}^*)^\mathsf{T}\mathbf{D},$$

This means that $\big(t^*\mathbf{a}^\mathsf{T} + (\mathbf{c}^*)^\mathsf{T}\mathbf{B}_{i^*} - (\mathbf{u}^*)^\mathsf{T}\big)\mathbf{D} = 0$. Since $t^*\mathbf{a}^\mathsf{T} + (\mathbf{c}^*)^\mathsf{T}\mathbf{B}_{i^*} \neq (\mathbf{u}^*)^\mathsf{T}$, this means that $g_1^{t^*\mathbf{a}^\mathsf{T} + (\mathbf{c}^*)^\mathsf{T}\mathbf{B}_{i^*} - (\mathbf{u}^*)^\mathsf{T}}$ is a valid solution to the k-KerLin challenge. Thus, $\mathcal{B}$ breaks k-KerLin with the same advantage ε. □

To complete the proof, it suffices to show that in Hyb_1, the output is 0 with overwhelming probability. We show that this is the case for all (possibly unbounded) adversaries $\mathcal{A}$. Our analysis will rely on the following claim from [KW15, Lemma 2]:

Lemma C.10 ([KW15, Lemma 2])**.** *Let n, t, k be integers. Fix any matrix $\mathbf{M} \in \mathbb{Z}_p^{n\times t}$ and $\mathbf{D} \in \mathbb{Z}_p^{(k+1)\times k}$. Then, for all (possibly unbounded) adversaries $\mathcal{A}$, if we sample $\mathbf{K} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{n\times(k+1)}$ and $(\mathbf{z}, \mathbf{y}) \leftarrow \mathcal{A}(\mathbf{M}^\mathsf{T}\mathbf{K}, \mathbf{K}\mathbf{D})$,*

$$\Pr[\mathbf{y} \notin \mathrm{span}(\mathbf{M}) \wedge \mathbf{z}^\mathsf{T} = \mathbf{y}^\mathsf{T}\mathbf{K}] \leq 1/p.$$

To complete the proof, let $\widetilde{\mathsf{crs}} = \big(\mathcal{G}, H, g_1^{\mathbf{V}}, g_2^{\hat{\mathbf{a}}}, g_2^{\mathbf{D}}, \{g_1^{\mathbf{w}_i}, g_1^{\mathbf{Z}_i}, g_2^{\hat{\mathbf{B}}_i}\}_{i\in[\rho]}\big)$ be the simulated CRS in Hyb_1, and let $(\sigma^*, i^*, r^*, \pi^*)$ be the adversary's output, where $\sigma^* = g_1^{(\mathbf{c}^*)^\mathsf{T}}$ and $\pi^* = (g_1^{t^*}, g_1^{(\mathbf{u}^*)^\mathsf{T}})$. The state $\mathsf{st}_\mathcal{E}$ is $\mathsf{st}_\mathcal{E} = (H, \mathbf{s}_1, \ldots, \mathbf{s}_k)$. In this case, $\mathbf{w}_i = \mathbf{V}\mathbf{s}_i$, $\hat{\mathbf{a}}^\mathsf{T} = \mathbf{a}^\mathsf{T}\mathbf{D}$, $\mathbf{Z}_i = \mathbf{w}_i\mathbf{a}^\mathsf{T} + \mathbf{V}\mathbf{B}_i$, and $\hat{\mathbf{B}}_i = \mathbf{B}_i\mathbf{D}$. We now show that with overwhelming probability, the output of Hyb_1 is 0:

- If $\mathsf{Verify}(\widetilde{\mathsf{crs}}, \sigma^*, i^*, r^*, \pi^*) = 0$, then the output in Hyb_1 is 0 by definition.
- Suppose $t^* = (\mathbf{c}^*)^\mathsf{T}\mathbf{s}_{i^*}$. Then, if $\mathsf{Verify}(\widetilde{\mathsf{crs}}, \sigma^*, i^*, r^*, \pi^*) = 1$, it must be the case that $r^* = H(g_1^{t^*}) = H(g_1^{(\mathbf{c}^*)^\mathsf{T}\mathbf{s}_{i^*}})$. Let $r \leftarrow \mathcal{E}_2(\mathsf{st}_\mathcal{E}, \sigma^*)$. By definition $r_i = H(g_1^{(\mathbf{c}^*)^\mathsf{T}\mathbf{s}_{i^*}})$. But in this case, $r^* = r_i$, and the output in Hyb_1 is 0.
- Suppose that $t^* \neq (\mathbf{c}^*)^\mathsf{T}\mathbf{s}_{i^*}$. Here, we will rely on Lemma C.10. Let $\mathbf{w} \in \mathbb{Z}_p^{\rho(\rho+k)}$ be the vector formed by stacking $\mathbf{w}_1, \ldots, \mathbf{w}_\rho$, and define matrices $\mathbf{K}$ and $\mathbf{M}$ as follows:

$$\mathbf{K} = \begin{bmatrix} \mathbf{a}^\mathsf{T} \\ \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_\rho \end{bmatrix} \in \mathbb{Z}_p^{(\rho k+1)\times(k+1)} \quad \text{and} \quad \mathbf{M}^\mathsf{T} = [\ \mathbf{w} \ \mid\ \mathbf{I}_\rho \otimes \mathbf{V}\ ] \in \mathbb{Z}_p^{\rho(\rho+k)\times(\rho k+1)}.$$

  By construction, observe that all of the group elements in $\widetilde{\mathsf{crs}}$ are a function of the components of $\mathbf{M}$ (i.e., the components $\mathbf{V}$, $\mathbf{w}_i$), $\mathbf{D}$, $\mathbf{M}^\mathsf{T}\mathbf{K}$ (i.e., $\mathbf{Z}_i$), and $\mathbf{K}\mathbf{D}$ (i.e., $\hat{\mathbf{a}}, \hat{\mathbf{B}}_i$). Thus, by Lemma C.10, over a uniform choice of $\mathbf{K}$ (correspondingly, over a random choice of the verification parameters $\mathbf{a}$, $\mathbf{B}_i$), no adversary $\mathcal{A}$ (given $\mathbf{M}$, $\mathbf{D}$, $\mathbf{M}^\mathsf{T}\mathbf{K}$, and $\mathbf{K}\mathbf{D}$) can output $\mathbf{y}$ and $\mathbf{u}^*$ such that $\mathbf{y} \notin \mathrm{span}(\mathbf{M})$ and $(\mathbf{u}^*)^\mathsf{T} = \mathbf{y}^\mathsf{T}\mathbf{K}$, except with negligible probability. Consider $\mathbf{y}^\mathsf{T} = [\ t^* \mid \mathbf{e}_{i^*}^\mathsf{T} \otimes (\mathbf{c}^*)^\mathsf{T}\ ] \in \mathbb{Z}_p^{\rho+1}$, where $\mathbf{e}_{i^*} \in \mathbb{Z}_p^\rho$ denotes the $(i^*)^{\text{th}}$ basis vector. Since $t^* \neq (\mathbf{c}^*)^\mathsf{T}\mathbf{s}_{i^*}$, $\mathbf{y} \notin \mathrm{span}(\mathbf{M})$, so by Lemma C.10, $\Pr[\mathbf{y}^\mathsf{T}\mathbf{K} = (\mathbf{u}^*)^\mathsf{T}] = \mathsf{negl}(\lambda)$. Since $\mathbf{y}^\mathsf{T}\mathbf{K} = t^*\mathbf{a}^\mathsf{T} + (\mathbf{c}^*)^\mathsf{T}\mathbf{B}_{i^*}$, this means that with overwhelming probability,

$$t^*\mathbf{a}^\mathsf{T} + (\mathbf{c}^*)^\mathsf{T}\mathbf{B}_{i^*} \neq (\mathbf{u}^*)^\mathsf{T}.$$

  In this case, if $\mathsf{Verify}(\widetilde{\mathsf{crs}}, \sigma^*, i^*, r^*, \pi^*) = 1$, then the output in Hyb_1 is 0 by definition. (And if the output of Verify is 0, the output in Hyb_1 is also 0). With overwhelming probability, the output of Hyb_1 in this case is 0. □

Proof of Theorem 4.17 (Statistical Simulation). Follows by the same argument as in the proof of Theorem 4.8. □

C.3 Analysis of Construction 4.18 (Dual-Mode (Malicious) HBG from k-Lin)

In this section, we give the proofs for the correctness and security theorems (Theorems 4.19 to 4.23) for the hidden-bits generator with malicious security from the k-Lin assumption (Construction 4.18).

Proof of Theorem 4.19 (Correctness). Follows by an analogous argument as in the proof of Theorem 4.4. □

Proof of Theorem 4.20 (Succinctness). The commitment σ in Construction 4.18 consists of a PRG seed $\mathsf{s} \in \{0,1\}^\kappa$ where $\kappa = \mathsf{poly}(\lambda)$ and $k = O(1)$ group elements $g^{\mathbf{c}^\mathsf{T}} \in \mathbb{G}^k$. Thus, $|\sigma| = \mathsf{poly}(\lambda)$. □

Proof of Theorem 4.21 (CRS Indistinguishability). Same as the proof of Theorem 4.6. □

Proof of Theorem 4.22 (Statistical Binding). This follows by an analogous argument as the proof of Theorem 4.7. In particular, we first define the following (inefficient) Open algorithm:

- $\mathsf{Open}(\mathsf{crs}, \sigma) \to r$: On input a $\mathsf{crs} = (\mathcal{G}, H, g^{\mathbf{V}}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_{\ell'}})$ and a commitment $\sigma = (\mathsf{s}, g^{\mathbf{c}^\mathsf{T}})$, the open algorithm first recovers $\mathbf{V}, \mathbf{w}_1, \ldots, \mathbf{w}_{\ell'}$ by solving the discrete logarithm problem over $\mathbb{G}$. For each $i \in [\ell']$, it checks that $\mathbf{w}_i = \mathbf{V}\mathbf{s}_i$ for some $\mathbf{s}_i \in \mathbb{Z}_p^k$, and outputs $\perp$ if not. If all checks pass, it computes $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$ and the shifted sets $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$ for each $i \in [\rho]$. It computes $r_i \leftarrow H\big(\prod_{j \in S_i} g^{\alpha_j \mathbf{c}^\mathsf{T} \mathbf{s}_j}\big)$ for each $i \in [\rho]$, and outputs r.

As in the proof of Theorem 4.7, we use a hybrid argument to complete the proof:

- Hyb_0: This is the real binding experiment.
- Hyb_1: Same as Hyb_0, except when generating the public key $\mathsf{pk} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_{\ell'}})$, the challenger samples $\mathbf{x}_1, \ldots, \mathbf{x}_{\ell'} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ and sets $\mathbf{z}_i \leftarrow \mathbf{V}\mathbf{x}_i$. Afterwards, it samples $a \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ and sets $\mathbf{b}_i \leftarrow \mathbf{x}_i - \mathbf{s}_i a$ for use as the (secret) verification coefficients.
- Hyb_2: Same as Hyb_1 except the challenger implements $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$ by computing $\mathsf{Verify}^*(\sigma, r_i, \pi_i, (\mathbf{s}_1, \ldots, \mathbf{s}_{\ell'}), (\mathbf{x}_1, \ldots, \mathbf{x}_{\ell'}))$, which does the following. On input $\sigma = (\mathsf{s}, g^{\mathbf{c}^\mathsf{T}})$, $r_i \in \{0,1\}$, $\pi_i = \{(j, g^{t_j}, g^{u_j})\}_{j \in S}$ for some implicitly-defined set $S \subseteq [\ell']$, and vectors $\mathbf{s}_1, \ldots, \mathbf{s}_{\ell'} \in \mathbb{Z}_p^k$, $\mathbf{x}_1, \ldots, \mathbf{x}_{\ell'} \in \mathbb{Z}_p^k$, compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$, and the shifted set $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$. It then checks the following conditions and outputs 1 only of them are satisfied (and outputs 0 otherwise):

$$S = S_i \quad \text{and} \quad r_i = H\big(\textstyle\prod_{j \in S_i} g^{\alpha_j t_j}\big) \quad \text{and} \quad \forall j \in S_i : t_j = \mathbf{c}^\mathsf{T}\mathbf{s}_j \text{ and } \mathbf{u}_j = \mathbf{c}^\mathsf{T}\mathbf{x}_j$$

As usual, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output of Hyb_i with adversary $\mathcal{A}$. Hybrids $\mathsf{Hyb}_0(\mathcal{A})$ and $\mathsf{Hyb}_1(\mathcal{A})$ are identical by construction. Hybrids $\mathsf{Hyb}_1(\mathcal{A})$ and $\mathsf{Hyb}_2(\mathcal{A})$ are statistically indistinguishable by a similar argument as that in the proof of Lemma C.3. Namely, the Verify algorithm in Hyb_1 outputs 1 only if for all $j \in S_i$:

$$u_j = t_j a + \mathbf{c}^\mathsf{T}\mathbf{b}_j = \mathbf{c}^\mathsf{T}(\mathbf{s}_j a + \mathbf{b}_j) + (t_j - \mathbf{c}^\mathsf{T}\mathbf{s}_j)a = \mathbf{c}^\mathsf{T}\mathbf{x}_j + (t_j - \mathbf{c}^\mathsf{T}\mathbf{s}_j)a.$$

If there is some $j \in S_i$ where $t_j \neq \mathbf{c}^\mathsf{T}\mathbf{s}_j$, then we can argue (as in the proof of Lemma C.3) that with overwhelming probability over the choice of a, the verifier in Hyb_1 outputs 0. Conversely, if $t_j = \mathbf{c}^\mathsf{T}\mathbf{s}_j$ for all $j \in S_i$, then Verify and Verify^* behave identically, and the claim follows. Finally, by a similar argument as in the proof of Theorem 4.7, the output of Hyb_2 is always 0. □

Proof of Theorem 4.23 (Statistical Simulation). We construct a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ as follows:

- $\mathcal{S}_1(1^\lambda, 1^\rho) \to (\mathsf{st}_\mathcal{S}, \widetilde{\mathsf{crs}})$: Sample $\mathcal{G} = (\mathbb{G}, p, g) \leftarrow \mathsf{GroupGen}(1^\lambda)$, $H \xleftarrow{\mathrm{R}} \mathcal{H}$, $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(\ell'+k)\times k}$, $\mathbf{w}_1, \ldots, \mathbf{w}_{\ell'} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell'+k}$ exactly as in $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$. In the following, we will write
$$\mathbf{W} = [\ \mathbf{w}_1 \mid \cdots \mid \mathbf{w}_{\ell'} \mid \mathbf{V}\ ] \in \mathbb{Z}_p^{(\ell'+k)\times(\ell'+k)}.$$
Output $\widetilde{\mathsf{crs}} = (\mathcal{G}, H, g^\mathbf{V}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_{\ell'}})$, and $\mathsf{st}_\mathcal{S} = (\widetilde{\mathsf{crs}}, \mathbf{W})$
- $\mathcal{S}_2(\mathsf{st}_\mathcal{S}, \mathsf{pk}, I, r_I) \to (\widetilde{\sigma}, \{\widetilde{\pi}_i\}_{i\in I})$: On input the simulation state $\mathsf{st}_\mathcal{S} = (\widetilde{\mathsf{crs}}, \mathbf{W})$ where $\widetilde{\mathsf{crs}} = (\mathcal{G}, H, g^\mathbf{V}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_{\ell'}})$, a public key $\mathsf{pk} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_{\ell'}})$, a set of indices $I \subseteq \{0,1\}^k$, and a bitstring $r_I \in \{0,1\}^{|I|}$, the simulator samples a seed $\mathsf{s} \xleftarrow{\mathrm{R}} \{0,1\}^\kappa$ and computes $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$. For each $i \in I$, it computes the shifted sets $S_i \leftarrow \{j + \ell\cdot(i-1) \mid j \in \hat{S}_i\}$. Then, it samples a vector $\mathbf{t} \in \mathbb{Z}_p^{\ell'}$ as follows:
  - For each $i \in I$, first sample $t'_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ subject to $H(g^{t'_i}) = r_i$. Specifically, the simulator repeatedly samples $t'_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ until finding one that satisfies $H(g^{t'_i}) = r_i$. If no such t'_i is found after λ iterations, then the simulator aborts and outputs $\perp$. Then for $j \in S_i$, sample $t_j \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ subject to the constraint that $\sum_{j\in S_i} \alpha_j t_j = t'_i$.
  - For all of the remaining indices $j \in [\ell'] \setminus \bigcup_{i\in I} S_i$, sample $t_j \xleftarrow{\mathrm{R}} \mathbb{Z}_p$.

  Next, it samples $\mathbf{c} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ and computes $\mathbf{y}^\mathsf{T} \leftarrow [\ \mathbf{t}^\mathsf{T} \mid \mathbf{c}^\mathsf{T}\ ] \cdot \mathbf{W}^{-1}$ (outputting $\perp$ if $\mathbf{W}$ is not invertible). It computes $g^{u_j} \leftarrow g^{\mathbf{y}^\mathsf{T}\mathbf{z}_j}$ for each $j \in S_i$ and $i \in I$. Finally, for each $i \in I$, it sets $\widetilde{\pi}_i = \{(j, g^{t_j}, g^{u_j})\}_{j\in S_i}$. Finally, it outputs $\widetilde{\sigma} = (\mathsf{s}, g^{\mathbf{c}^\mathsf{T}})$ and $\{\widetilde{\pi}_i\}_{i\in I}$.

To show that $\mathsf{ExptHide}[\mathcal{A}, 0]$ and $\mathsf{ExptHide}[\mathcal{A}, 1]$ are statistically indistinguishable, we use a hybrid argument:

- Hyb_0: This is the distribution in $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 0]$. Namely, the challenger begins by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$. It gives crs to $\mathcal{A}$ to receive a public key pk. For each challenge query, the challenger samples $(\sigma, r, \{\pi_i\}_{i\in[\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs}, \mathsf{pk})$ and gives r to $\mathcal{A}$ to receive a set $I \subseteq [\rho]$. It then replies with σ and $\{\pi_i\}_{i\in I}$.
- Hyb_1: Same as Hyb_0 except the challenger computes $(\mathsf{st}_\mathcal{S}, \widetilde{\mathsf{crs}}) \leftarrow \mathcal{S}_1(1^\lambda, 1^\rho)$ and uses $\widetilde{\mathsf{crs}}$ in place of crs. Everything else proceeds identically to Hyb_0. Specifically, in this experiment, the challenger samples $\mathcal{G}$, H, and $\mathbf{V}, \mathbf{w}_1, \ldots, \mathbf{w}_{\ell'}$ as specified by $\mathcal{S}_1$ and sets $\widetilde{\mathsf{crs}} = (\mathcal{G}, H, g^\mathbf{V}, g^{\mathbf{w}_1}, \ldots, g^{\mathbf{w}_{\ell'}})$. It gives $\widetilde{\mathsf{crs}}$ to the adversary and receives a public key $\mathsf{pk} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_{\ell'}})$. Let $\mathbf{Z} \in \mathbb{Z}_p^{(\ell'+k)\times\ell'}$ be the matrix whose columns are $\mathbf{z}_1, \ldots, \mathbf{z}_{\ell'}$. On a challenge query, the challenger proceeds as follows:
  1. Sample $\mathsf{s} \xleftarrow{\mathrm{R}} \{0,1\}^\kappa$ and compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$. For each $i \in [\rho]$, compute $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$.

2. Sample $\mathbf{y} \xleftarrow{R} \mathbb{Z}_p^{\ell'+k}$ and compute $[\ \mathbf{t}^\mathsf{T} \mid \mathbf{c}^\mathsf{T}\ ] \leftarrow \mathbf{y}^\mathsf{T}\mathbf{W}$, $\mathbf{u}^\mathsf{T} \leftarrow \mathbf{y}^\mathsf{T}\mathbf{Z}$.
3. For each $i \in [\rho]$, compute $t'_i \leftarrow \prod_{j \in S_i} g^{\alpha_j t_j}$, $r_i \leftarrow H(t'_i)$ and $\pi_i \leftarrow \{(j, g^{t_j}, g^{u_j})\}_{j \in S_i}$.
4. The challenger gives $r \in \{0,1\}^\rho$ to $\mathcal{A}$ and receives a set $I \subseteq [\rho]$.
5. The challenger replies with $\sigma = (\mathsf{s}, g^{t_{\ell'+1}}) = (\mathsf{s}, g^{\mathbf{c}^\mathsf{T}})$ and the set $\{\pi_i\}_{i \in I}$.

- Hyb_2: Same as Hyb_1, except when responding to the challenge queries, instead of sampling $\mathbf{y} \xleftarrow{R} \mathbb{Z}_p^{\ell'+k}$, the challenger instead samples $\mathbf{t} \xleftarrow{R} \mathbb{Z}_p^{\ell'}$ and $\mathbf{c} \xleftarrow{R} \mathbb{Z}_p^{k}$. It sets $\mathbf{y}^\mathsf{T} \leftarrow [\ \mathbf{t}^\mathsf{T} \mid \mathbf{c}^\mathsf{T}\ ] \cdot \mathbf{W}^{-1}$ (and outputs $\perp$ if $\mathbf{W}$ is not invertible). All remaining components are constructed as in Hyb_1.

- Hyb_3: Same as Hyb_2, except when responding to the challenge queries, the challenger first samples $\mathbf{t}' \xleftarrow{R} \mathbb{Z}_p^{\rho}$. Next, for each $i \in [\rho]$ and $j \in S_i$, it samples $t_j \xleftarrow{R} \mathbb{Z}_p$ subject to the constraint that $\sum_{j \in S_i} \alpha_j t_j = t'_i$. For indices $j \in [\ell'] \setminus \bigcup_{i \in [\rho]} S_i$, sample $t_j \xleftarrow{R} \mathbb{Z}_p$.

- Hyb_4: Same as Hyb_3, except when responding to the challenge queries, the challenger first samples $r \xleftarrow{R} \{0,1\}^\rho$. Then, for each $i \in [\rho]$, it samples $t'_i \xleftarrow{R} \mathbb{Z}_p$ conditioned on $H(t'_i) = r_i$. All remaining components are constructed as in Hyb_3.

- Hyb_5: Same as Hyb_4 except when responding to the challenge queries, the challenger samples $\mathbf{t} \in \mathbb{Z}_p^{\ell'}$ *after* it receives the challenge set. In particular, on each query, after the challenger receives the set $I \subseteq [\rho]$, it samples $\mathbf{t}$ as follows:

  – For each $i \in I$, first sample $t'_i \xleftarrow{R} \mathbb{Z}_p$ subject to $H(g^{t'_i}) = r_i$. Then, for $j \in S_i$, sample $t_j \xleftarrow{R} \mathbb{Z}_p$ subject to the constraint that $\sum_{j \in S_i} \alpha_j t_j = t'_i$.
  – For all of the remaining indices $j \in [\ell'+1] \setminus \bigcup_{i \in I} S_i$, sample $t_j \xleftarrow{R} \mathbb{Z}_p$.

  After sampling $\mathbf{t}$ using the above procedure, the simulator constructs the remaining components as in Hyb_4. This is the distribution in $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 1]$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output of hybrid Hyb_i with adversary $\mathcal{A}$. In the following, we will show that the output distribution of each pair of adjacent experiments is statistically indistinguishable (or identically distributed). Our analysis will rely on the following *statistical* property on the output of a secure PRG:

Claim C.11. *Suppose G is a secure PRG, and take $\mathsf{s} \xleftarrow{R} \{0,1\}^\kappa$. Let $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$. Then, with overwhelming probability over the choice of s, the following properties hold:*

- *For any $i \in [\rho\ell]$, $\Pr[\alpha_i = 0] = \mathsf{negl}(\lambda)$.*
- *Let $\mathbf{A} \in \mathbb{Z}_p^{\ell \times \ell'}$ be any fixed matrix and let $I \subseteq [\rho]$ be any fixed set of indices. For each $i \in I$, let $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$ be the shifted set of indices, and define*

$$J = \{j \in S_i \text{ for some } i \in I \mid j \in [\ell']\} \subseteq [\ell'].$$

  Let $\hat{\mathbf{A}} \in \mathbb{Z}_p^{\ell \times \lambda |I|}$ be the submatrix of $\mathbf{A}$ formed by taking only the columns of $\mathbf{A}$ indexed by J. For $i \in [\rho]$, define $\boldsymbol{\alpha}^{(i)} \in \mathbb{Z}_p^{\ell}$ as follows:

$$\alpha_j^{(i)} = \begin{cases} \alpha_{j+\ell\cdot(i-1)} & \text{if } j \in \hat{S}_i \\ 0 & \text{otherwise.} \end{cases}$$

By construction, $\boldsymbol{\alpha}^{(i)}$ contains exactly λ non-zero entries (as specified by the set $\hat{S}_i$). Then, for any $i \in [\rho] \setminus I$,

$$\Pr[\boldsymbol{\alpha}^{(i)} \in \mathrm{span}(\hat{\mathbf{A}})] = \Pr[\exists \mathbf{v} \in \mathbb{Z}_p^{\lambda|I|} : \boldsymbol{\alpha}^{(i)} = \hat{\mathbf{A}}\mathbf{v}] = \mathsf{negl}(\lambda).$$

Proof. Both properties above are efficiently-checkable given $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha})$, the matrix $\mathbf{A}$, and the set of indices I. It suffices to show that both hold with overwhelming probability when $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \xleftarrow{\mathrm{R}} \mathcal{T}_{\lambda,\ell}^{\rho} \times \mathbb{Z}_p^{\rho\ell}$. The claim then follows by PRG security. We show each property below:

- Since $\boldsymbol{\alpha} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\rho\ell}$, $\alpha_i = 0$ with probability $1/p = \mathsf{negl}(\lambda)$.
- Let $\hat{\mathbf{A}}$ be the matrix defined above, and let $\hat{\mathbf{a}}_j^\mathsf{T} \in \mathbb{Z}_p^{\lambda|I|}$ denote the j^{th} row of $\hat{\mathbf{A}}$. Let $n = \mathrm{rank}(\hat{\mathbf{A}}) \leq \lambda\,|I| \leq \lambda\rho = \ell/3$. This means that there exists a collection of indices $j_1, \ldots, j_n \in [\ell]$ such that the collection $\{\hat{\mathbf{a}}_{j_1}^\mathsf{T}, \ldots, \hat{\mathbf{a}}_{j_n}^\mathsf{T}\}$ is linearly independent and $\mathrm{span}(\hat{\mathbf{A}}^\mathsf{T}) = \mathrm{span}(\{\hat{\mathbf{a}}_{j_1}^\mathsf{T}, \ldots, \hat{\mathbf{a}}_{j_n}^\mathsf{T}\})$. By construction, $\hat{\mathbf{A}}$ only depends on $\mathbf{A}$ and the sets $\hat{S}_i$ where $i \in I$. This means that we can sample the sets $\hat{S}_i$ where $i \notin I$ *after* fixing $\hat{\mathbf{A}}$ and the set of indices $\{j_1, \ldots, j_n\}$. In this case, since $\hat{S}_i \xleftarrow{\mathrm{R}} \mathcal{T}_{\lambda,\ell}$,

$$\Pr[\hat{S}_i \xleftarrow{\mathrm{R}} \mathcal{T}_{\lambda,\ell} : \hat{S}_i \subseteq \{j_1, \ldots, j_n\}] \leq \frac{\binom{n}{\lambda}}{\binom{\ell}{\lambda}} \leq \left(\frac{ne}{\ell}\right)^\lambda = \mathsf{negl}(\lambda),$$

since $n \leq \ell/3$. Thus, for any $i \in [\rho]\setminus I$, there exists an index $j^* \in \hat{S}_i$ where $j^* \notin \{j_1, \ldots, j_n\}$ with overwhelming probability. In addition, since $\hat{\mathbf{a}}_{j^\star}^\mathsf{T} \in \mathrm{span}(\hat{\mathbf{A}}^\mathsf{T})$, there exist scalars $\beta_1, \ldots, \beta_n \in \mathbb{Z}_p$ such that $\hat{\mathbf{a}}_{j^*}^\mathsf{T} = \sum_{\tau\in[n]} \beta_\tau \hat{\mathbf{a}}_{j_\tau}^\mathsf{T}$. This means that if there exists $\mathbf{v} \in \mathbb{Z}_p^{\lambda|I|}$ such that $\boldsymbol{\alpha}^{(i)} = \hat{\mathbf{A}}\mathbf{v}$, then

$$\alpha_{j^*}^{(i)} = \hat{\mathbf{a}}_{j^*}^\mathsf{T}\mathbf{v} = \sum_{\tau\in[n]} \beta_\tau \hat{\mathbf{a}}_{j_\tau}^\mathsf{T}\mathbf{v} = \sum_{\tau\in[n]} \beta_\tau \alpha_{j_\tau}^{(i)}.$$

Since $j^* \in \hat{S}_i$, by construction, $\alpha_{j^*}^{(i)} = \alpha_{j^*+\ell\cdot(i-1)}$, which is uniform over $\mathbb{Z}_p$ and independent of β_τ and $\alpha_{j_\tau}^{(i)}$ for all $\tau \in [n]$. Thus, over the randomness of $\boldsymbol{\alpha}$, for any $i \in [\rho] \setminus I$

$$\Pr[\exists \mathbf{v} \in \mathbb{Z}_p^{\lambda|I|} : \boldsymbol{\alpha}^{(i)} = \hat{\mathbf{A}}\mathbf{v}] = \Pr\left[\alpha_{j^*}^{(i)} = \sum_{\tau\in[n]} \beta_\tau \alpha_{j_\tau}^{(i)}\right] = \frac{1}{p} = \mathsf{negl}(\lambda). \qquad \square$$

We will also use the following simple fact on linear independence.

Claim C.12. *Take any matrix $\mathbf{A} \in \mathbb{Z}_p^{m\times n}$ and any vector $\mathbf{v} \in \mathbb{Z}_p^m$ where $\mathbf{v} \notin \mathrm{span}(\mathbf{A})$. Then, for any $\gamma \in \mathbb{Z}_p$, the following distributions are identical:*

$$\{\mathbf{t} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^m : (\mathbf{A}, \mathbf{v}, \gamma, \mathbf{t}^\mathsf{T}\mathbf{A})\} \equiv \{\mathbf{t} \xleftarrow{\mathrm{R}} \{\mathbf{t} \in \mathbb{Z}_p^m \mid \mathbf{t}^\mathsf{T}\mathbf{v} = \gamma\} : (\mathbf{A}, \mathbf{v}, \gamma, \mathbf{t}^\mathsf{T}\mathbf{A})\}$$

Proof. Follows immediately from the fact that $\mathbf{v}$ is not in the span of $\mathbf{A}$. $\square$

Lemma C.13. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \equiv \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. Since $\mathcal{S}_1(1^\lambda, 1^\rho)$ samples $\widetilde{\mathsf{crs}}$ using the same procedure as $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$, the output distributions of Hyb_0 and Hyb_1 are identically distributed. □

Lemma C.14. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. With overwhelming probability, $\mathbf{W}$ is full-rank and invertible. In this case, the distribution of $\mathbf{y}$ in Hyb_1 and Hyb_2 is identical. □

Lemma C.15. *If G is a secure PRG, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_2(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_3(\mathcal{A})$.*

Proof. These two distributions are identical provided that for each $i \in [\rho]$, there is at least one index $j \in S_i$ where $\alpha_j \neq 0$. In this case, both $\mathbf{t}$ and $\mathbf{t}'$ are uniformly random over $\mathbb{Z}_p^{\ell'}$ and $\mathbb{Z}_p^{\rho}$, respectively, subject to the condition that $t'_i = \sum_{j \in S_i} \alpha_j t_j$ for all $i \in [\rho]$. By Claim C.11, with overwhelming probability, $\alpha_j \neq 0$ for any $j \in [\ell']$. Since $\rho = \mathsf{poly}(\lambda)$, the claim follows by a union bound. □

Lemma C.16. *If $\mathcal{H}$ is statistically uniform, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_3(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_4(\mathcal{A})$.*

Proof. In Hyb_3, the challenger samples $t'_i \stackrel{\mathrm{R}}{\leftarrow} \mathbb{Z}_p$ and sets $r_i \leftarrow H(t'_i)$ while, in Hyb_4, the challenger samples $r_i \stackrel{\mathrm{R}}{\leftarrow} \{0,1\}$ and sets $t'_i \stackrel{\mathrm{R}}{\leftarrow} \mathbb{Z}_p$ subject to $H(t'_i) = r_i$. These two distributions are statistically indistinguishable by the argument from the proof of Lemma C.6. □

Lemma C.17. *If G is a secure PRG, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_4(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_5(\mathcal{A})$.*

Proof. The challenger samples $\widetilde{\mathsf{crs}}$ identically in the two experiments, so it suffices to consider its responses to the challenge queries. Let $\mathbf{W} \in \mathbb{Z}_p^{(\ell'+k)\times(\ell'+k)}$ be the matrix of components in the CRS and let $\mathsf{pk} = (g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_{\ell'}})$ be the adversary's chosen public key. As above, let $\mathbf{Z} \in \mathbb{Z}_p^{(\ell'+k)\times\ell'}$ be the matrix whose columns are $\mathbf{z}_1, \ldots, \mathbf{z}_{\ell'}$. Now, on each challenge query, in both Hyb_4 and Hyb_5, the challenger starts by sending the adversary a random string $r \stackrel{\mathrm{R}}{\leftarrow} \{0,1\}^\rho$, and the adversary replies with a set $I \subseteq [\rho]$. The challenger then replies with a commitment $\sigma = (\mathsf{s}, g^{\mathbf{c}^\mathsf{T}})$ and a set of proofs $\{\pi_i\}_{i \in I}$, where $\pi_i = \{(j, g^{t_j}, g^{u_j})\}_{j \in S_i}$. To complete the proof, it suffices to show that the challenger's response is statistically indistinguishable in the two experiments. To facilitate the analysis, we first define the following variables:

- For $i \in [\rho]$, let $\mathbf{t}^{(i)} \in \mathbb{Z}_p^{\ell}$ be the vector where $t_j^{(i)} = t_{j+\ell\cdot(i-1)}$. In other words, $\mathbf{t}^\mathsf{T} = [\ (\mathbf{t}^{(1)})^\mathsf{T} \mid \cdots \mid (\mathbf{t}^{(\rho)})^\mathsf{T}\ ] \in \mathbb{Z}_p^{\ell'}$.
- For $i \in [\rho]$, define $\boldsymbol{\alpha}^{(i)} \in \mathbb{Z}_p^{\ell}$ as follows:

$$\alpha_j^{(i)} = \begin{cases} \alpha_{j+\ell\cdot(i-1)} & \text{if } j \in \hat{S}_i \\ 0 & \text{otherwise.} \end{cases}$$

  By construction, in both experiments, $t'_i = \sum_{j \in [\ell]} \alpha_j^{(i)} t_j^{(i)}$.

We now consider the different components in the two experiments:

- The string r is independently and uniformly sampled from $\{0,1\}^\rho$ in both experiments.
- The seed s is independently and uniformly sampled from $\{0,1\}^\kappa$ in both experiments, so the seed s, the sets $S_1, \ldots, S_\rho$, and the vector $\boldsymbol{\alpha}$ are identically distributed in the two experiments.

- The commitment vector $\mathbf{c}$ is independently and uniformly sampled from $\mathbb{Z}_p^k$ in both experiments.
- For $i \in I$, t'_i is sampled identically in the two experiments (namely, $t'_i \xleftarrow{R} \mathbb{Z}_p$ subject to $H(g^{t'_i}) = r_i$). Correspondingly, the vectors $\mathbf{t}^{(i)}$ for all $i \in I$ are identically distributed.

It suffices to show that in Hyb_4 and Hyb_5, the components $\{u_j\}_{j \in S_i}$ for all $i \in I$ are statistically indistinguishable (given all of the other components in the adversary's view). In both experiments,

$$\mathbf{u}^\mathsf{T} = \mathbf{y}^\mathsf{T}\mathbf{Z} = [\ \mathbf{t}^\mathsf{T} \mid \mathbf{c}^\mathsf{T}\ ] \cdot \mathbf{W}^{-1}\mathbf{Z}.$$

Let

$$\mathbf{A} = \mathbf{W}^{-1}\mathbf{Z} = \begin{bmatrix} \mathbf{A}^{(1)} \\ \vdots \\ \mathbf{A}^{(\rho)} \\ \mathbf{A}' \end{bmatrix} \in \mathbb{Z}_p^{(\ell'+k)\times\ell'},$$

where $\mathbf{A}^{(i)} \in \mathbb{Z}_p^{\ell\times\ell'}$ and $\mathbf{A}' \in \mathbb{Z}_p^{k\times\ell'}$. This is a *fixed* matrix (determined by the CRS and the adversary's choice of public key). Moreover, the adversary's choice of indices $i \in I$ is *fixed* before the challenger samples the commitment and the openings. Let

$$J = \{j \in S_i \text{ for some } i \in I \mid j \in [\ell']\}$$

be the set of indices that appear in some set S_i for $i \in I$. Let $\hat{\mathbf{A}} \in \mathbb{Z}_p^{(\ell'+k)\times\lambda|I|}$ be the submatrix of $\mathbf{A}$ formed by taking only the columns of $\mathbf{A}$ indexed by the set J. In particular, the elements of $[\ \mathbf{t}^\mathsf{T} \mid \mathbf{c}^\mathsf{T}\ ] \cdot \hat{\mathbf{A}}$ precisely coincide with $\bigcup_{i\in I}\{u_j\}_{j\in S_i}$, which are the elements in the adversary's view. Now, write $\hat{\mathbf{A}}$ as follows:

$$\hat{\mathbf{A}} = \begin{bmatrix} \hat{\mathbf{A}}^{(1)} \\ \vdots \\ \hat{\mathbf{A}}^{(\rho)} \\ \hat{\mathbf{A}}' \end{bmatrix},$$

where $\hat{\mathbf{A}}^{(i)} \in \mathbb{Z}_p^{\ell\times\lambda|I|}$ and $\hat{\mathbf{A}}' \in \mathbb{Z}_p^{k\times\lambda|I|}$. With this, we can write

$$[\ \mathbf{t}^\mathsf{T} \mid \mathbf{c}^\mathsf{T}\ ] \cdot \hat{\mathbf{A}} = \sum_{i\in[\rho]} \left((\mathbf{t}^{(i)})^\mathsf{T}\hat{\mathbf{A}}^{(i)}\right) + \mathbf{c}^\mathsf{T}\hat{\mathbf{A}}' \in \mathbb{Z}_p^{\lambda|I|} \tag{C.2}$$

Since $\mathbf{c}$ and all of the $\mathbf{t}^{(i)}$ are sampled independently, we can consider each summand individually:

- If $i \in I$, then as argued above, $\mathbf{t}^{(i)}$ is identically distributed in Hyb_4 and Hyb_5. Since $\hat{\mathbf{A}}$ is a fixed matrix (independent of $\mathbf{t}$), the products $(\mathbf{t}^{(i)})^\mathsf{T}\hat{\mathbf{A}}^{(i)}$ are also identically distributed.
- If $i \notin I$, then the $\mathbf{t}^{(i)}$ in the two experiments are sampled from different distributions. In Hyb_4, $\mathbf{t}^{(i)}$ is uniform over $\mathbb{Z}_p^\ell$ subject to $\sum_{j\in[\ell]} \alpha_j^{(i)} t_j^{(i)} = t'_i$ while, in Hyb_5, $\mathbf{t}^{(i)}$ is uniform over $\mathbb{Z}_p^\ell$.

  Since $i \notin I$ and $\mathbf{A}^{(i)} \in \mathbb{Z}_p^{\ell\times\ell'}$ is a fixed matrix, we can appeal to Claim C.11 to conclude that the probability that $\boldsymbol{\alpha}^{(i)} \in \mathrm{span}(\hat{\mathbf{A}}^{(i)})$ is negligible. By Claim C.12, this means that with overwhelming probability, the distribution of $\left(\mathbf{t}^{(i)}\right)^\mathsf{T}\hat{\mathbf{A}}^{(i)}$ is statistically indistinguishable in Hyb_4 (where $\mathbf{t}^{(i)}$ is uniform subject to a linear constraint $\boldsymbol{\alpha}^{(i)} \notin \mathrm{span}(\hat{\mathbf{A}}^{(i)})$) and Hyb_5 (where $\mathbf{t}^{(i)}$ is uniform).

- As argued above, $\mathbf{c}$ is identically distributed in the two experiments (and independent of $\hat{\mathbf{A}}$). Thus, $\mathbf{c}^\mathsf{T}\hat{\mathbf{A}}'$ is identically distributed in the two experiments.

Since every term in Eq. (C.2) is either statistically indistinguishable or identically distributed in the two experiments, we conclude that $[\ \mathbf{t}^\mathsf{T} \mid \mathbf{c}^\mathsf{T}\ ] \cdot \hat{\mathbf{A}}$ is also statistically indistinguishable. Based on the above analysis, the adversary's view in each challenge query is statistically indistinguishable in Hyb_4 and Hyb_5. Since the adversary makes at most a polynomial number of challenge queries, the claim follows by a hybrid argument. □

Since each consecutive pair of hybrid experiments is statistically indistinguishable (or identically distributed), the claim follows. □

D Analysis of Constructions from QR (Section 5)

In this section, we provide the analysis of the dual-mode hidden-bits generators from Section 5.

D.1 Analysis of Construction 5.3 (Dual-Mode HBG from QR)

In this section, we give the proofs for the correctness and security theorems (Theorems 5.4 to 5.8) for the dual-mode hidden-bits generator from QR (Construction 5.3).

Proof of Theorem 5.4 (Correctness). Fix $\lambda \in \mathbb{N}$, a polynomial $\rho = \rho(\lambda)$, an index $i \in [\rho]$, and a mode $\mathsf{mode} \in \{\mathsf{binding}, \mathsf{hiding}\}$. Let $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode})$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$, and $(\sigma, r, \{\pi_i\}_{i\in[\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs}, \mathsf{pk})$. By construction, $\mathsf{crs} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1\mathbf{v}}h^{\hat{\mathbf{w}}_1}, \ldots, g^{s_\rho\mathbf{v}}h^{\hat{\mathbf{w}}_\rho})$, for some vectors $\mathbf{v} \in \mathbb{Z}^\rho_{p'q'}$ and $\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_\rho \in \mathbb{Z}^\rho_2$. Then $\mathsf{sk} = \{a_\tau, b_{\tau,i}\}_{\tau\in[T],i\in[\rho]}$ and

$$\mathsf{pk} = \{\overline{\mathbf{z}}_{\tau,i}\}_{\tau\in[T],i\in[\rho]} = \{(g^{s_i\mathbf{v}}h^{\hat{\mathbf{w}}_i})^{a_\tau}(g^{\mathbf{v}})^{b_{\tau,i}}\}_{\tau\in[T],i\in[\rho]} = \{g^{(a_\tau s_i + b_{\tau,i})\mathbf{v}}h^{a_\tau\hat{\mathbf{w}}_i}\}_{\tau\in[T],i\in[\rho]}.$$

Next, we have that $\sigma = \overline{c} = \prod_{j\in[\rho]} g^{v_j y_j} = g^{\mathbf{y}^\mathsf{T}\mathbf{v}}$, $r_i = \mathsf{LEQ}(\overline{t}_i, \overline{t}_i h)$ and $\pi_i = (\overline{t}_i, u_i)$, where

$$\overline{t}_i = \prod_{j\in[\rho]} (g^{s_i v_j}h^{\hat{w}_{i,j}})^{y_j} = g^{s_i\mathbf{y}^\mathsf{T}\mathbf{v}}h^{\mathbf{y}^\mathsf{T}\hat{\mathbf{w}}_i},$$

and $u_i = H(\overline{u}_{1,i}, \ldots, \overline{u}_{T,i})$ where

$$\overline{u}_{\tau,i} = \prod_{j\in[\rho]} \left(g^{(a_\tau s_i + b_{\tau,i}v_j)}h^{a_\tau\hat{w}_{i,j}}\right)^{y_j} = g^{(a_\tau s_i + b_{\tau,i})\mathbf{y}^\mathsf{T}\mathbf{v}}h^{a_\tau\mathbf{y}^\mathsf{T}\hat{\mathbf{w}}_i}$$

for all $\tau \in [T]$. Consider now the behavior of $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$. By construction, $r_i = \mathsf{LEQ}(\overline{t}_i, \overline{t}_i h)$, so it suffices to check that Eq. (5.1) holds. By construction, for each $\tau \in [T]$

$$\overline{t}_\tau^{a_\tau}\overline{c}^{b_{\tau,i}} = (g^{a_\tau s_i\mathbf{y}^\mathsf{T}\mathbf{v}}h^{a_\tau\mathbf{y}^\mathsf{T}\hat{\mathbf{w}}_i})(g^{b_{\tau,i}\mathbf{y}^\mathsf{T}\mathbf{v}}) = g^{(a_\tau s_i + b_{\tau,i})\mathbf{y}^\mathsf{T}\mathbf{v}}h^{a_\tau\mathbf{y}^\mathsf{T}\hat{\mathbf{w}}_i} = \overline{u}_{\tau,i}.$$

Thus, Eq. (5.1) is equivalent to checking whether $u_i = H(\overline{u}_{1,i}, \ldots, \overline{u}_{T,i})$, which holds by construction, and Verify outputs 1. □

Proof of Theorem 5.5 (Succinctness). The size of the commitment in Construction 5.3 consists of a single element in $\mathbb{Z}^*_N$, which has length $\lceil \log N \rceil = \mathsf{poly}(\lambda)$. □

Proof of Theorem 5.6 (CRS Indistinguishability). The proof of Theorem 5.6 follows via the following claim from [BG10]:

Claim D.1 ([BG10, §5]). *Suppose a safe prime product modulus sampler* $\mathsf{SampleModulus}$ *satisfies the* QR *assumption. Let* $h = -1$*. Then, for all polynomials* $\rho = \rho(\lambda)$*, all fixed vectors* $\hat{\mathbf{w}} \in \{0,1\}^\rho$*, and all efficient adversaries* $\mathcal{A}$*, if we sample* $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$*,* $\mathbf{v} \xleftarrow{R} \mathbb{Z}_{p'q'}^\rho$*,* $s \xleftarrow{R} \mathbb{Z}_{p'q'}$*, where* $p = 2p' + 1$ *and* $q = 2q' + 1$*, we have that*

$$\left|\Pr[\mathcal{A}(N, g, g^{\mathbf{v}}, g^{s\mathbf{v}}) = 1] - \Pr[\mathcal{A}(N, g, g^{\mathbf{v}}, g^{s\mathbf{v}} h^{\hat{\mathbf{w}}}) = 1]\right| = \mathsf{negl}(\lambda),$$

where g *is a generator of* $\mathbb{QR}_N$*.*

First, Claim D.1 also holds if we instead sample $\mathbf{v} \xleftarrow{R} \mathbb{Z}_{\lfloor N/2 \rfloor}^\rho$ and $s \xleftarrow{R} \mathbb{Z}_{\lfloor N/2 \rfloor}$ since the statistical distance between $\{r \xleftarrow{R} \mathbb{Z}_{\lfloor N/2 \rfloor} : r \bmod p'q'\}$ and $\mathsf{Uniform}(\mathbb{Z}_{p'q'})$ is

$$\Delta(\{r \xleftarrow{R} \mathbb{Z}_{\lfloor N/2 \rfloor} : r \bmod p'q'\}, \mathsf{Uniform}(\mathbb{Z}_{p'q'})) = \frac{(N-1)/2 \bmod p'q'}{(N-1)/2} = \frac{p' + q'}{2p'q' + p' + q'} = \mathsf{negl}(\lambda),$$

since $1/p', 1/q' = \mathsf{negl}(\lambda)$. To complete the proof, we use a hybrid argument, where for $j \in \{0, \ldots, \rho\}$, Hyb_j is defined as follows:

- Hyb_j: Sample $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$, $\mathbf{v} \xleftarrow{R} \mathbb{Z}_{\lfloor N/2 \rfloor}^\rho$, $s_1, \ldots, s_\rho \xleftarrow{R} \mathbb{Z}_{\lfloor N/2 \rfloor}$, $H \xleftarrow{R} \mathcal{H}$, and let g be a generator of $\mathbb{QR}_N$. Output $\mathsf{crs} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1\mathbf{v}}, \ldots, g^{s_i\mathbf{v}}, g^{s_{i+1}\mathbf{v}} h^{\mathbf{e}_{i+1}}, \ldots, g^{s_\rho \mathbf{v}} h^{\mathbf{e}_\rho})$.

By construction, Hyb_0 implements $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$ and Hyb_ρ implements $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{binding})$. We can appeal to Claim D.1 (where $\mathbf{v}$ and s are sampled from $\mathbb{Z}_{\lfloor N/2 \rfloor}^\rho$ and $\mathbb{Z}_{\lfloor N/2 \rfloor}$, respectively) to argue that each pair of adjacent hybrids Hyb_{j-1} and Hyb_j for $j \in [\rho]$ are computationally indistinguishable. To see this, set $\hat{\mathbf{w}} = \mathbf{e}_j$ in Claim D.1, and let $(N, g, g^{\mathbf{v}}, g^{s\mathbf{v}} h^{\hat{\mathbf{w}}'})$ be the challenge. We simulate a CRS by sampling $s_1, \ldots, s_{j-1}, s_{j+1}, \ldots, s_\rho \xleftarrow{R} \mathbb{Z}_{\lfloor N/2 \rfloor}$, $H \xleftarrow{R} \mathcal{H}$ and outputting

$$(N, g, h, H, g^{\mathbf{v}}, g^{s_1\mathbf{v}}, \ldots, g^{s_{j-1}\mathbf{v}}, g^{s\mathbf{v}} h^{\hat{\mathbf{w}}'}, g^{s_{j+1}\mathbf{v}} h^{\mathbf{e}_{j+1}}, \ldots, g^{s_\rho \mathbf{v}} h^{\mathbf{e}_\rho}).$$

If $\hat{\mathbf{w}}' = \mathbf{0}$, then this is precisely the distribution Hyb_j and if $\hat{\mathbf{w}}' = \mathbf{e}_j$, then this is the distribution Hyb_{j-1}. Thus, by Claim D.1, the outputs of Hyb_{j-1} and Hyb_j are computationally indistinguishable for all $j \in [\rho]$. Since $\rho = \mathsf{poly}(\lambda)$, Theorem 5.6 follows by a hybrid argument. □

Proof of Theorem 5.7 (Statistical Binding). Recall that in binding mode, the common reference string is given by

$$\mathsf{crs} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1\mathbf{v}}, \ldots, g^{s_\rho \mathbf{v}}).$$

We define the (inefficient) Open algorithm as follows:

- $\mathsf{Open}(\mathsf{crs}, \sigma) \to r$: On input a $\mathsf{crs} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1\mathbf{v}}, \ldots, g^{s_\rho \mathbf{v}})$ and a commitment $\sigma = g^c$ (outputting $\perp$ if the components do not have this form), the open algorithm recovers $\mathbf{v}$, $s_1, \ldots, s_\rho$, and c. Then, it computes $r_i \leftarrow \mathsf{LEQ}(g^{cs_i}, g^{cs_i} h)$ for each $i \in [\rho]$, and outputs r.

To complete the proof, we use a hybrid argument:

- Hyb_0: This is the real soundness experiment. The challenger samples $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{binding})$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$ and gives $(\mathsf{crs}, \mathsf{pk})$ to $\mathcal{A}$. Here,

$$\mathsf{crs} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1\mathbf{v}}, \ldots, g^{s_\rho \mathbf{v}}) \quad \text{and} \quad \mathsf{pk} = \{g^{(a_\tau s_i + b_{\tau,i})\mathbf{v}}\}_{\tau \in [T], i \in [\rho]}.$$

  The adversary can make queries to the verification oracle, and on each query (σ, i, r_i, π_i), the challenger replies with $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$. At the end of the game, the adversary outputs a tuple $(\sigma^*, i^*, r^*, \pi^*)$ and the output of the experiment is 1 if $r^* \neq r_i$ where $r \leftarrow \mathsf{Open}(\mathsf{crs}, \sigma^*)$ and $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma^*, i^*, r^*, \pi^*) = 1$.

- Hyb_1: Same as Hyb_0 except the challenger samples the scalars $s_1, \ldots, s_\rho$ and the secret key components $a_\tau, b_{\tau,i}$ uniformly at random from $\mathbb{Z}_{2p'q'}$ (instead of $\mathbb{Z}_{\lfloor N/2 \rfloor}$) for all $\tau \in [T]$ and $i \in [\rho]$.

- Hyb_2: Same as Hyb_1 except the challenger computes $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$ using the following modified procedure. First, parse $\sigma = g^c h^{\hat{c}}$ and $\pi_i = (g^{t_i} h^{\hat{t}_i}, u_i)$, for some $c, t_i \in \mathbb{Z}_{p'q'}$, $\hat{c}, \hat{t}_i \in \mathbb{Z}_2$, and $u_i \in \{0,1\}^\lambda$. Then, the challenger does the following:

  - If $r_i \neq \mathsf{LEQ}(g^{t_i} h^{\hat{t}_i}, g^{t_i} h^{\hat{t}_i + 1})$, output 0.
  - If $\hat{c} \neq 0$ or $\hat{t}_i \neq 0$, then output 0.
  - If $t_i \neq s_i c$, then output 0.
  - Otherwise, take any $\mathbf{y} \in \mathbb{Z}_{p'q'}^\rho$ such that $\mathbf{y}^\mathsf{T}\mathbf{v} = c$. Write the public key as $\mathsf{pk} = \{g^{\mathbf{z}_{\tau,i}}\}_{\tau \in [T], i \in [\rho]}$ and output 1 if $u_i = H\big(g^{\mathbf{y}^\mathsf{T}\mathbf{z}_{1,i}}, \ldots, g^{\mathbf{y}^\mathsf{T}\mathbf{z}_{T,i}}\big)$. Otherwise (or if no such $\mathbf{y}$ exists), output 0.

  Importantly, the challenger's responses to the verification queries in Hyb_2 depend *only* on the public components (i.e., crs and pk).

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of an execution of experiment Hyb_i with adversary $\mathcal{A}$. We now show that the output distribution of each adjacent pair of hybrid experiments is statistically indistinguishable.

Lemma D.2. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. The statistical distance between $\mathsf{Uniform}(\mathbb{Z}_{\lfloor N/2 \rfloor})$ and $\mathsf{Uniform}(\mathbb{Z}_{2p'q'})$ satisfies

$$\Delta(\mathsf{Uniform}(\mathbb{Z}_{\lfloor N/2 \rfloor}), \mathsf{Uniform}(\mathbb{Z}_{2p'q'})) = 1 - \frac{2p'q'}{\lfloor N/2 \rfloor} = 1 - \frac{2p'q'}{2p'q' + p' + q'} = \mathsf{negl}(\lambda), \tag{D.1}$$

since $1/p', 1/q' = \mathsf{negl}(\lambda)$. Since $\rho, T = \mathsf{poly}(\lambda)$, the claim follows by a union bound. □

Lemma D.3. *If $\mathcal{H}$ is pairwise independent, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. In Hyb_1 and Hyb_2, the challenger evaluates Verify at most $Q+1$ times where $Q = \mathsf{poly}(\lambda)$ is the bound on the number of queries the adversary makes. For $j \in \{0, \ldots, Q+1\}$, let $\mathsf{Hyb}_{1,j}$ denote the experiment where the first j queries are handling according to the specification in Hyb_2 while the remaining queries are handling according to the specification in Hyb_1. By construction, $\mathsf{Hyb}_1 \equiv \mathsf{Hyb}_{1,0}$ and $\mathsf{Hyb}_2 \equiv \mathsf{Hyb}_{1,Q+1}$. Consider $\mathsf{Hyb}_{1,j-1}$ and $\mathsf{Hyb}_{1,j}$ for $j \in [Q+1]$. These two

experiments only differ in how the challenger computes the output for the j^{th} Verify call. Moreover, by construction of $\mathsf{Hyb}_{1,j-1}$ and $\mathsf{Hyb}_{1,j}$, all of the adversary's queries prior to the j^{th} query are handled according to the specification in Hyb_2, which depend only on the public components crs and pk. Let (σ, i, r_i, π_i) be the arguments to the j^{th} Verify call. Write $\sigma = g^c h^{\hat{c}}$ and $\pi_i = (g^{t_i} h^{\hat{t}_i}, u_i)$ for $c, t_i \in \mathbb{Z}_{p'q'}$, $\hat{c}, \hat{t}_i \in \mathbb{Z}_2$, and $u_i \in \{0,1\}^\lambda$. We consider each case individually:

- Suppose $r_i \neq \mathsf{LEQ}(g^{t_i} h^{\hat{t}_i}, g^{t_i} h^{\hat{t}_i + 1})$. Then, the output in both experiments is 0.
- Suppose $\hat{c} \neq 0$ or $\hat{t}_i \neq 0$. We argue that with overwhelming probability, the output in $\mathsf{Hyb}_{1,j-1}$ is 0. For the output to be 1 in $\mathsf{Hyb}_{1,j-1}$, it must be the case that $u_i = H(\boldsymbol{\omega}_i)$ where

$$\boldsymbol{\omega}_i = \left((g^{t_i} h^{\hat{t}_i})^{a_1} (g^c h^{\hat{c}})^{b_{1,i}}, \ldots, (g^{t_i} h^{\hat{t}_i})^{a_T} (g^c h^{\hat{c}})^{b_{T,i}}\right) \in \mathbb{J}_N^T. \tag{D.2}$$

  In both Hyb_1 and Hyb_2, the public parameters are *independent* of the values of a_τ mod 2 and $b_{\tau,i}$ mod 2 for all $\tau \in [T]$. This follows from the fact that $a_\tau, b_{\tau,i} \in \mathbb{Z}_{2p'q'}$, and the adversary only sees elements $g^{(a_\tau s_i + b_{\tau,i})\mathbf{v}}$, where g generates a group of order $p'q'$ (and $\gcd(p'q', 2) = 1$). Next, since a_τ and $b_{\tau,i}$ are uniform over $\mathbb{Z}_{2p'q'}$, the values a_τ mod 2 and $b_{\tau,i}$ mod 2 are distributed uniformly and independently of the rest of the public parameters. Since the responses to all of the adversary's queries prior to its j^{th} query only depend on the public parameters, the conditional distribution of a_τ mod 2 and $b_{\tau,i}$ mod 2 given the adversary's view up to the time of its j^{th} query is uniform and independently random. Since at least one of $\hat{c}$ and $\hat{t}_i$ is non-zero, this means that the value of $a_\tau \hat{t}_i + b_{\tau,i}\hat{c}$ mod 2 is independently and uniformly random over $\mathbb{Z}_2$, and correspondingly, the conditional min-entropy of the vector $\boldsymbol{\omega}_i$ from Eq. (D.2) is at least $T = 2(\lambda + \lceil \log N \rceil)$. We now appeal to Lemma 2.6 to argue that the output distribution of $H(\boldsymbol{\omega}_i)$ is $(3p'q'\varepsilon)$-close to uniform over $\{0,1\}^\lambda$ where

$$\varepsilon = 2^{-(T-\lambda)/2} = 2^{-\lambda/2 - \lceil \log N \rceil},$$

  since the adversary can choose the values of $c, t_i \in \mathbb{Z}_{p'q'}$ and $(\hat{c}, \hat{t}_i) \in \mathbb{Z}_2^2 \setminus \{\mathbf{0}\}$ *after* seeing the hash function H. Since $3p'q' < N$, $3p'q'\varepsilon < 2^{-\lambda/2} = \mathsf{negl}(\lambda)$, so we conclude that conditioned on the adversary's view, the distribution of $H(\boldsymbol{\omega}_i)$ in $\mathsf{Hyb}_{1,j-1}$ is statistically close to uniform over $\{0,1\}^\lambda$. The probability that $u_i = H(\boldsymbol{\omega}_i)$ is then negligibly close to $1/2^\lambda$, and the challenger outputs 0 with overwhelming probability.
- Suppose that $t_i \neq s_i c \in \mathbb{Z}_{p'q'}$. We only need to consider the case where $\hat{c} = 0$ and $\hat{t}_i = 0$. We show that in this case, the output in $\mathsf{Hyb}_{1,j-1}$ is 0 with overwhelming probability. For the output to be 1 in $\mathsf{Hyb}_{1,j-1}$ in this case, we require that $u_i = H(\boldsymbol{\omega}_i)$ where

$$\boldsymbol{\omega}_i = \left(g^{a_1 t_i + b_{1,i} c}, \ldots, g^{a_T t_i + b_{T,i} c}\right) \in \mathbb{QR}_N^T. \tag{D.3}$$

  The only components in crs and pk that depend on a_τ and $b_{\tau,i}$ for $\tau \in [T]$ are the public-key components $g^{a_\tau s_i \mathbf{v} + b_{\tau,i}\mathbf{v}}$. Let $\mathbf{b}_\tau \in \mathbb{Z}_{2p'q'}^\rho$ be the vector whose components are $b_{\tau,1}, \ldots, b_{\tau,\rho}$, and let $\mathbf{s} \in \mathbb{Z}_{2p'q'}^\rho$ be the vector whose components are $s_1, \ldots, s_\rho$. The public parameters pk can then be expressed as a function of

$$\mathbf{Z}_\tau = [\ \mathbf{s} \mid \mathbf{I}_\rho\ ] \cdot \begin{bmatrix} a_\tau \\ \mathbf{b}_\tau \end{bmatrix} \cdot \mathbf{v}^\mathsf{T} \in \mathbb{Z}_{2p'q'}^{\rho \times \rho},$$

where $\mathbf{I}_\rho \in \mathbb{Z}_{2p'q'}^{\rho\times\rho}$ is the identity matrix. Namely, the components of pk consist of $g^{\mathbf{Z}_\tau}$ for all $\tau \in [T]$. Since $t_i \neq s_i c \in \mathbb{Z}_{p'q'}$, by the Chinese remainder theorem, it must be the case that $t_i \neq s_i c \pmod{p'}$ or $t_i \neq s_i c \pmod{q'}$. Without loss of generality, suppose that $t_i \neq s_i c \pmod{p'}$. In this case the vector $[\ t_i \mid c\cdot\mathbf{e}_i\ ]$ is linearly independent of the rows of the matrix $[\ \mathbf{s} \pmod{p'} \mid \mathbf{I}_\rho\ ]$. Since $a_\tau, \mathbf{b}_\tau$ are uniform over $\mathbb{Z}_{2p'q'}$, the components $a_\tau \bmod p'$ and $\mathbf{b}_\tau \bmod p'$ are uniform over $\mathbb{Z}_{p'}$. By linear independence over $\mathbb{Z}_{p'}$, the value of $a_\tau t_i + b_{\tau,i}c \pmod{p'}$ is uniformly random over $\mathbb{Z}_{p'}$ even given $\mathbf{Z}_\tau$ for all $\tau \in [T]$. As such, the conditional min-entropy of the vector $\boldsymbol{\omega}_i$ from Eq. (D.3) is at least $T \log p' > T$. By an analogous argument as in the previous case, we can now appeal to Lemma 2.6 and conclude that the output distribution of $H(\boldsymbol{\omega}_i)$ is statistically close to uniform over $\{0,1\}^\lambda$, in which case the probability that $u_i = H(\boldsymbol{\omega}_i)$ is negligible.

- The only remaining case is when $\hat{c} = 0 = \hat{t}_i$, $t_i = s_i c$, and $r_i = \mathsf{LEQ}(g^{t_i}, g^{t_i} h)$. In this case, $(g^{t_i})^{a_\tau}(g^c)^{b_{\tau,i}} = g^{(a_\tau s_i + b_{\tau,i})c}$ for all $\tau \in [T]$. Since $\mathbf{v} \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N/2 \rfloor}$, with overwhelming probability, there will be some component that is invertible modulo $p'q'$. If so, there always exists $\mathbf{y} \in \mathbb{Z}_{p'q'}$ such that $\mathbf{y}^\mathsf{T}\mathbf{v} = c$. Then, in $\mathsf{Hyb}_{1,j-1}$, the challenger outputs 1 if and only if

$$\begin{aligned} u_i = H\big(g^{(a_1 s_i + b_{1,i})c}, \ldots, g^{(a_T s_i + b_{T,i})c}\big) &= H\big(g^{(a_1 s_i + b_{1,i})\mathbf{y}^\mathsf{T}\mathbf{v}}, \ldots, g^{(a_T s_i + b_{T,i})\mathbf{y}^\mathsf{T}\mathbf{v}}\big) \\ &= H\big(g^{\mathbf{y}^\mathsf{T}\mathbf{z}_1}, \ldots, g^{\mathbf{y}^\mathsf{T}\mathbf{z}_T}\big), \end{aligned}$$

  where $\mathbf{z}_{\tau,i} = (a_\tau s_i + b_{\tau,i})\mathbf{v}$ are the components in the public key. This is precisely the verification relation in $\mathsf{Hyb}_{1,j}$.

In each case, we see that the j^{th} call to Verify is implemented correctly with overwhelming probability in $\mathsf{Hyb}_{1,j-1}$ and $\mathsf{Hyb}_{1,j}$. □

To complete the proof, it suffices to show that for all adversaries $\mathcal{A}$, the output of $\mathsf{Hyb}_2(\mathcal{A})$ is 0 with overwhelming probability. Let $(\sigma^*, i^*, r^*, \pi^*)$ be the adversary's output in Hyb_2. The output of Hyb_2 is 1 only if $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma^*, i^*, r^*, \pi^*) = 1$ and $r_{i^*} \neq r^*$ where $r \leftarrow \mathsf{Open}(\mathsf{crs}, \sigma^*)$. Write $\sigma^* = g^c h^{\hat{c}}$ and $\pi^* = (g^{t_{i^*}} h^{\hat{t}_{i^*}}, u_{i^*})$ In Hyb_2, if $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma^*, i^*, r^*, \pi^*) = 1$, it must be the case that $\hat{c} = 0$, $\hat{t}_{i^*} = 0$, $t_{i^*} = s_{i^*} c$, and

$$r^* = \mathsf{LEQ}\big(g^{t_{i^*}} h^{\hat{t}_{i^*}}, g^{t_{i^*}} h^{\hat{t}_{i^*}+1}\big) = \mathsf{LEQ}\big(g^{s_{i^*}c}, g^{s_{i^*}c} h\big).$$

Moreover, since $\hat{c} = 0$, $r_{i^*} = \mathsf{LEQ}(g^{cs_{i^*}}, g^{cs_{i^*}} h) = r^*$. In this case, the output in Hyb_2 is 0, and the theorem follows. □

Proof of Theorem 5.8 (Statistical Simulation). We construct a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ as follows:

- $\mathcal{S}_1(1^\lambda, 1^\rho) \to (\mathsf{st}_\mathcal{S}, \widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}})$: Sample $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$, where $p = 2p' + 1$ and $q = 2q' + 1$. Let g be a generator of $\mathbb{QR}_N$ and $h = -1$. Sample a vector $\mathbf{v} \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N/2 \rfloor}^\rho$, scalars $s_1, \ldots, s_\rho \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N/2 \rfloor}$, and a hash function $H \xleftarrow{\mathrm{R}} \mathcal{H}$. Set $\widetilde{\mathsf{crs}} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\mathbf{e}_1}, \ldots, g^{s_\rho \mathbf{v}} h^{\mathbf{e}_\rho})$. Sample $a_\tau, b_{\tau,i} \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N/2 \rfloor}$ for all $\tau \in [T]$ and $i \in [\rho]$. Output $\widetilde{\mathsf{crs}}$, $\widetilde{\mathsf{pk}} = \{g^{(a_\tau s_i + b_{\tau,i})\mathbf{v}} h^{a_\tau \mathbf{e}_i}\}_{\tau\in[T], i\in[\rho]}$, $\widetilde{\mathsf{sk}} = \{a_\tau, b_{\tau,i}\}_{\tau\in[T], i\in[\rho]}$, and $\mathsf{st}_\mathcal{S} = (\widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, p', q', s_1, \ldots, s_\rho)$.

- $\mathcal{S}_2(\mathsf{st}_{\mathcal{S}}, I, r_I) \to (\widetilde{\sigma}, \{\widetilde{\pi}_i\}_{i\in I})$: On input $\mathsf{st}_{\mathcal{S}} = (\widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, p', q', s_1, \ldots, s_\rho)$ where

$$\widetilde{\mathsf{crs}} = \big(N, g, h, H, g^{\mathbf{v}}, g^{s_1\mathbf{v}}h^{\mathbf{e}_1}, \ldots, g^{s_\rho\mathbf{v}}h^{\mathbf{e}_\rho}\big) \quad \text{and} \quad \widetilde{\mathsf{pk}} = \{g^{\mathbf{z}_{\tau,i}}h^{\hat{\mathbf{z}}_{\tau,i}}\}_{\tau\in[T],i\in[\rho]},$$

  a set of indices $I \subseteq [\rho]$, and a bitstring $r_I \in \{0,1\}^{|I|}$, the simulator samples $\mathbf{y}' \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{p'q'}$. Then, for each $i \in [\rho]$, it samples $\hat{y}'_i \in \mathbb{Z}_2$ as follows:

  - If $i \notin I$, sample $\hat{y}'_i \xleftarrow{\mathrm{R}} \mathbb{Z}_2$.
  - If $i \in I$, set $\hat{y}_i \in \mathbb{Z}_2$ to be the unique value where $r_i = \mathsf{LEQ}(g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}}h^{\hat{y}_i}, g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}}h^{\hat{y}_i+1})$.

  Define $\mathbf{y} \in \mathbb{Z}^{\rho}_{2p'q'}$ to be the vector where $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod 2$. The simulator then sets $g^{t_i}g^{\hat{t}_i} = g^{s_i\mathbf{y}^\mathsf{T}\mathbf{v}}h^{\hat{y}_i}$ and

$$u_i = H\big(g^{\mathbf{y}^\mathsf{T}\mathbf{z}_{1,i}}h^{\mathbf{y}^\mathsf{T}\hat{\mathbf{z}}_{1,i}}, \ldots, g^{\mathbf{y}^\mathsf{T}\mathbf{z}_{T,i}}h^{\mathbf{y}^\mathsf{T}\hat{\mathbf{z}}_{T,i}}\big).$$

  Output $\widetilde{\sigma} = g^{\mathbf{y}^\mathsf{T}\mathbf{v}}$ and $\{\widetilde{\pi}_i\}_{i\in I}$ where $\widetilde{\pi}_i = (g^{t_i}h^{\hat{t}_i}, u_i)$.

To show that $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 0]$ and $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 1]$ are statistically indistinguishable, we use a hybrid argument:

- Hyb_0: This is the experiment $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 0]$. Namely, the challenger begins by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$. For each challenge query, the challenger first samples $(\sigma, r, \{\pi_i\}_{i\in[\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs})$ and gives r to $\mathcal{A}$ before receiving a set $I \subseteq [\rho]$ chosen by $\mathcal{A}$. It then replies with σ and $\{\pi_i\}_{i\in I}$.

  More precisely, in this experiment,

$$\mathsf{crs} = \big(N, g, h, H, g^{\mathbf{v}}, g^{s_1\mathbf{v}}h^{\mathbf{e}_1}, \ldots, g^{s_\rho\mathbf{v}}h^{\mathbf{e}_\rho}\big) \quad \text{and} \quad \mathsf{pk} = \{g^{\mathbf{z}_{\tau,i}}h^{\hat{\mathbf{z}}_{\tau,i}}\}_{\tau\in[T],i\in[\rho]},$$

  where $\mathbf{z}_{\tau,i} = (a_\tau s_i + b_{\tau,i})\mathbf{v}$ and $\hat{\mathbf{z}}_{\tau,i} = a_\tau\mathbf{e}_i$. On each challenge query, the challenger samples $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N/2\rfloor}$ and computes

$$\sigma = g^{\mathbf{y}^\mathsf{T}\mathbf{v}} \quad \text{and} \quad g^{t_i}h^{\hat{t}_i} = g^{s_i\mathbf{y}^\mathsf{T}\mathbf{v}}h^{\mathbf{y}^\mathsf{T}\mathbf{e}_i} \quad \text{and} \quad g^{u_{\tau,i}}h^{\hat{u}_{\tau,i}} = g^{\mathbf{y}^\mathsf{T}\mathbf{z}_{\tau,i}}h^{\mathbf{y}^\mathsf{T}\hat{\mathbf{z}}_{\tau,i}},$$

  for all $\tau \in [T]$ and $i \in [\rho]$. The random bits r_i satisfy $r_i = \mathsf{LEQ}(g^{t_i}h^{\hat{t}_i}, g^{t_i}h^{\hat{t}_i+1})$. Finally, the proofs π_i are given by $\pi_i = (g^{t_i}h^{\hat{t}_i}, u_i)$, where $u_i = H(g^{u_{1,i}}h^{\hat{u}_{1,i}}, \ldots, g^{u_{T,i}}h^{\hat{u}_{T,i}})$.

- Hyb_1: Same as Hyb_0, except that the challenger computes $(\mathsf{st}_{\mathcal{S}}, \widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}}) \leftarrow \mathcal{S}_1(1^\lambda, 1^\rho)$ and uses $\widetilde{\mathsf{crs}}$, $\widetilde{\mathsf{pk}}$, and $\widetilde{\mathsf{sk}}$ instead of crs, pk, and sk, respectively.

- Hyb_2: Same as Hyb_1, except when responding to the challenge queries, the challenger samples $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{2p'q'}$ instead of $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{\lfloor N/2\rfloor}$.

- Hyb_3: Same as Hyb_2, except when responding to the challenge queries, the challenger first samples $\mathbf{y}' \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{p'q'}$ and $\hat{\mathbf{y}} \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_2$. It defines $\mathbf{y} \in \mathbb{Z}^{\rho}_{2p'q'}$ to be the vector where $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod 2$.

- Hyb_4: Same as Hyb_3, except when responding to the challenge queries, the challenger first samples $r \xleftarrow{R} \{0,1\}^\rho$. Then it samples $\mathbf{y}' \xleftarrow{R} \mathbb{Z}^\rho_{p'q'}$. For each $i \in [\rho]$, it sets $\hat{y}'_i \in \mathbb{Z}_2$ to be the unique value where $r_i = \mathsf{LEQ}(g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i}, g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i+1})$. Finally it sets $\mathbf{y} \in \mathbb{Z}^\rho_{2p'q'}$ to be the vector where $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod 2$. The remaining components are constructed as before.

- Hyb_5: Same as Hyb_4, except when responding to the challenge queries, the challenger samples $\hat{\mathbf{y}}' \in \mathbb{Z}^\rho_2$ *after* it receives the challenge set. In particular, on each query, after the challenger receives the set $I \subseteq [\rho]$, it sets $\hat{y}'_i \in \mathbb{Z}_2$ to the unique value where $r_i = \mathsf{LEQ}(g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i}, g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i+1})$ if $i \in I$, and otherwise, it samples $\hat{y}'_i \xleftarrow{R} \mathbb{Z}_2$. All remaining components are constructed as in Hyb_4. This is the distribution in $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 1]$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of an execution of experiment Hyb_i with adversary $\mathcal{A}$. We now show that the output distribution of each adjacent pair of hybrid experiments is statistically indistinguishable.

Lemma D.4. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \equiv \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. Since $\mathcal{S}_1(1^\lambda, 1^\rho)$ samples $\widetilde{\mathsf{crs}}$, $\widetilde{\mathsf{pk}}$, and $\widetilde{\mathsf{sk}}$ using the same procedure as Setup and KeyGen, the output distributions of hybrids Hyb_0 and Hyb_1 are identically distributed. □

Lemma D.5. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. The only difference between Hyb_1 and Hyb_2 is that the challenger samples $\mathbf{y}$ uniformly at random from $\mathbb{Z}^\rho_{2p'q'}$ instead of $\mathbb{Z}^\rho_{\lfloor N/2 \rfloor}$ when answering challenge queries. Since the distributions $\mathsf{Uniform}(\mathbb{Z}_{\lfloor N/2 \rfloor})$ and $\mathsf{Uniform}(\mathbb{Z}_{2p'q'})$ are statistically indistinguishable (see Eq. (D.1)), the claim follows by a union bound (since $\rho = \mathsf{poly}(\lambda)$ and $\mathcal{A}$ makes a polynomial number of queries). □

Lemma D.6. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_2(\mathcal{A}) \equiv \mathsf{Hyb}_3(\mathcal{A})$.*

Proof. The two distribution only differ in how $\mathbf{y}$ is sampled. In Hyb_2, $\mathbf{y}$ is uniform over $\mathbb{Z}^\rho_{2p'q'}$ while in Hyb_3, we sample $\mathbf{y}' \xleftarrow{R} \mathbb{Z}_{p'q'}$ and $\hat{\mathbf{y}}' \xleftarrow{R} \mathbb{Z}^\rho_2$ and define $\mathbf{y}$ so that $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod 2$. These distributions are identical by the Chinese remainder theorem (since $\gcd(p'q', 2) = 1$). □

Lemma D.7. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_3(\mathcal{A}) \equiv \mathsf{Hyb}_4(\mathcal{A})$.*

Proof. First, $\mathbf{y}'$ is sampled identically in the two distributions. It suffices to show that once we fix $\mathbf{y}'$, there is a one-to-one correspondence between the value of $\hat{\mathbf{y}}' \in \mathbb{Z}^\rho_2$ and the value of r. Since g generates a subgroup of order $p'q'$, and $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$, we have that in Hyb_3, $g^{s_i \mathbf{y}^\mathsf{T}\mathbf{v}} = g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}}$. Similarly, since h has order 2, $h^{\mathbf{y}^\mathsf{T}\mathbf{e}_i} = h^{y_i} = h^{\hat{y}'_i}$. This means that the bit r_i satisfies

$$r_i = \mathsf{LEQ}(g^{t_i} h^{\hat{t}_i}, g^{t_i} h^{\hat{t}_i+1}) = \mathsf{LEQ}(g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i}, g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i+1}).$$

By construction of LEQ and the fact that h generates a subgroup of order 2, this means that

$$\mathsf{LEQ}(g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{(1-\hat{y}'_i)}, g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{(1-\hat{y}'_i)+1}) = 1 - \mathsf{LEQ}(g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i}, g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i+1}) = 1 - r_i,$$

and so we see that there is a one-to-one correspondence between $\hat{\mathbf{y}}' \in \mathbb{Z}^\rho_2$ and the bitstring $r \in \{0,1\}^\rho$. Both Hyb_3 and Hyb_4 enforce this correspondence, and thus, are identically distributed. □

Lemma D.8. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_4(\mathcal{A}) \equiv \mathsf{Hyb}_5(\mathcal{A})$.*

Proof. First, the challenger generates the common reference string crs and the public key pk identically in the two experiments. It suffices to argue that the challenger's response to the adversary's challenge query is identically distributed in Hyb_4 and Hyb_5. In particular, on each challenge query, after the adversary outputs a set $I \subseteq [\rho]$, the challenger replies with a commitment $\sigma = g^{\mathbf{y}^\mathsf{T}\mathbf{v}}$ and a collection of proofs $\{\pi_i\}_{i \in I}$ where $\pi_i = (g^{t_i} h^{\hat{t}_i}, u_i)$. We show that the commitment σ and the components t_i, $\hat{t}_i$, and u_i are identically distributed in the two experiments:

- In Hyb_4 and Hyb_5, the challenger samples $\mathbf{y}' \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{p'q'}$. Since g generates a subgroup of order $p'q'$ and $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$, it follows that $\sigma = g^{\mathbf{y}^\mathsf{T}\mathbf{v}} = g^{(\mathbf{y}')^\mathsf{T}\mathbf{v}}$, and so σ is identically distributed in the two experiments.
- In Hyb_4 and Hyb_5, for all $i \in I$, the challenger samples $\hat{y}'_i \xleftarrow{\mathrm{R}} \mathbb{Z}_2$ conditioned on $r_i = \mathsf{LEQ}(g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}_i}, g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}_i+1})$. Thus, the variables $\hat{y}'_i$ for $i \in I$ in the two experiments are identically distributed. In both experiments, it then defines

  $$g^{t_i} g^{\hat{t}_i} = g^{s_i \mathbf{y}^\mathsf{T}\mathbf{v}} h^{\mathbf{y}^\mathsf{T}\mathbf{e}_i} = g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i},$$

  and so all of the $g^{t_i} h^{\hat{t}_i}$ terms for $i \in I$ are identically distributed.
- In Hyb_4 and Hyb_5, $u_i = H(g^{u_{1,i}} h^{\hat{u}_{1,i}}, \ldots, g^{u_{T,i}} h^{\hat{u}_{T,i}})$ where

  $$g^{u_{\tau,i}} h^{\hat{u}_{\tau,i}} = g^{\mathbf{y}^\mathsf{T}\mathbf{z}_{\tau,i}} h^{\mathbf{y}^\mathsf{T}\hat{\mathbf{z}}_{\tau,i}} = g^{(\mathbf{y}')^\mathsf{T}\mathbf{z}_{\tau,i}} h^{a_\tau \hat{y}'_i}, \tag{D.4}$$

  using the fact that g generates a group of order $p'q'$, h generates a group of order 2 and $\hat{\mathbf{z}}_{\tau,i} = a_\tau \mathbf{e}_i$. Since all of the components in Eq. (D.4) are identically distributed for all $i \in I$ and $\tau \in [T]$, we conclude that the components u_i for $i \in I$ are also identically distributed. □

Theorem 5.8 now follows by a hybrid argument. □

D.2 Analysis of Construction 5.9 (Dual-Mode (Malicious) HBG from QR)

In this section, we give the proofs for the correctness and security theorems (Theorems 5.10 to 5.14) for the dual-mode hidden-bits generator with malicious security from QR (Construction 5.9). The analysis proceeds very similarly to that of the basic QR scheme (Construction 5.3) in Appendix D.1.

Proof of Theorem 5.10 (Correctness). Follows by an analogous argument as the proof of Theorem 5.4. □

Proof of Theorem 5.11 (Succinctness). The commitment σ in Construction 5.9 consists of a PRG seed $\mathsf{s} \in \{0,1\}^\kappa$ where $\kappa = \mathsf{poly}(\lambda)$ and an element of $\mathbb{Z}_N$, which has size $2\lceil \log N \rceil = \mathsf{poly}(\lambda)$. Thus, $|\sigma| = \mathsf{poly}(\lambda)$. □

Proof of Theorem 5.12 (CRS Indistinguishability). Same as the proof of Theorem 5.6. □

Proof of Theorem 5.13 (Statistical Binding). Follows by a similar argument as in the proof of Theorem 5.7. Specifically, we first show that we can substitute the real verification algorithm Verify with the following algorithm. On input (σ, i, r_i, π_i), the challenger does the following:

- Write $\sigma = (\mathsf{s}, \overline{c})$ and $\pi_i = \{(j, \overline{t}_j, u_j)\}_{j \in S}$ for an implicitly defined set S. Let $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$, and define the shifted set $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$. If $S \neq S_i$, output 0.
- If $\overline{c} \neq g^c$ for some $c \in \mathbb{Z}_{p'q'}$, output 0.
- If $\overline{t}_j \neq g^{cs_j}$ for any $j \in S_i$, output 0. Here, $s_j \in \mathbb{Z}_{p'q'}$ is sampled by Setup and used to construct $\mathsf{crs} = (N, g, h, g^{\mathbf{v}}, g^{s_1 \mathbf{v}}, \ldots, g^{s_{\ell'} \mathbf{v}})$
- Compute $\overline{t}'_i \leftarrow \prod_{j \in S_i} \overline{t}_j^{\alpha_j}$ and output 0 if $r_i \neq \mathsf{LEQ}(\overline{t}'_i, \overline{t}'_i h)$.
- Take any $\mathbf{y} \in \mathbb{Z}^{\ell'}_{p'q'}$ such that $\mathbf{y}^\mathsf{T} \mathbf{v} = c$, and write the public key as $\mathsf{pk} = \{g^{\mathbf{z}_{\tau,i}}\}_{\tau \in [T], i \in [\ell']}$. If $u_j \neq H(g^{\mathbf{y}^\mathsf{T} \mathbf{z}_{1,j}}, \ldots, g^{\mathbf{y}^\mathsf{T} \mathbf{z}_{T,j}})$ for some $j \in S_i$, or if no such $\mathbf{y}$ exists, then output 0. If all checks pass, output 1.

Using the same argument as the proof of Theorem 5.7 (applied to each individual component σ, $\overline{t}_j$, and u_j), and appealing to a union bound, we can conclude that the output of the real verification algorithm and this modified verification algorithm are identical with overwhelming probability. Similarly, as in the proof of Theorem 5.7, no adversary is able to win the binding game with respect to the modified verification algorithm, and the claim follows. □

Proof of Theorem 5.14 (Statistical Hiding). We construct a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ as follows:

- $\mathcal{S}_1(1^\lambda, 1^\rho) \to (\mathsf{st}_\mathcal{S}, \widetilde{\mathsf{crs}})$: Sample $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$, where $p = 2p' + 1$ and $q = 2q' + 1$. Let g be a generator of $\mathbb{QR}_N$ and $h = -1$ be the generator of $\mathbb{H} = \{\pm 1\}$. Sample a vector $\mathbf{v} \xleftarrow{\mathrm{R}} \mathbb{Z}^{\ell'}_{\lfloor N/2 \rfloor}$, scalars $s_1, \ldots, s_{\ell'} \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N/2 \rfloor}$, and a hash function $H \xleftarrow{\mathrm{R}} \mathcal{H}$. Output

$$\widetilde{\mathsf{crs}} = \left(N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\mathbf{e}_1}, \ldots, g^{s_{\ell'} \mathbf{v}} h^{\mathbf{e}_{\ell'}}\right).$$

  and $\mathsf{st}_\mathcal{S} = (\widetilde{\mathsf{crs}}, p', q', s_1, \ldots, s_{\ell'})$.

- $\mathcal{S}_2(\mathsf{st}_\mathcal{S}, \mathsf{pk}, I, r_I) \to (\widetilde{\sigma}, \{\widetilde{\pi}_i\}_{i \in I})$: On input $\mathsf{st}_\mathcal{S} = (\widetilde{\mathsf{crs}}, p', q', s_1, \ldots, s_{\ell'})$ where

$$\widetilde{\mathsf{crs}} = \left(N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\mathbf{e}_1}, \ldots, g^{s_{\ell'} \mathbf{v}} h^{\mathbf{e}_{\ell'}}\right),$$

  a public key $\mathsf{pk} = \{\overline{\mathbf{z}}_{\tau,i}\}_{\tau \in [T], i \in [\ell']}$, a set of indices $I \subseteq [\rho]$, and a bitstring $r_I \in \{0,1\}^{|I|}$, the simulator does the following:

  1. Check that $\overline{\mathbf{z}}_{\tau,i} \in \mathbb{J}^{\ell'}_N$ for all $\tau \in [T]$ and $i \in [\ell']$. Output $\perp$ if this is not the case.
  2. Sample a seed $\mathsf{s} \xleftarrow{\mathrm{R}} \{0,1\}^\kappa$ and compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$, where $\boldsymbol{\alpha} \in \mathbb{Z}_2^{\rho \ell}$. For each $i \in I$, it computes the shifted sets $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$.
  3. Sample $\mathbf{y}' \xleftarrow{\mathrm{R}} \mathbb{Z}^{\ell'}_{p'q'}$. Then, it samples a vector $\hat{\mathbf{y}}' \in \mathbb{Z}^{\ell'}_2$ as follows:
     - For each $i \in I$, let $\omega_i \leftarrow \sum_{j \in S_i} \alpha_j s_j \pmod{p'q'}$. Then, set $\hat{\omega}_i \in \mathbb{Z}_2$ to be the unique value where $r_i = \mathsf{LEQ}(g^{\omega_i (\mathbf{y}')^\mathsf{T} \mathbf{v}} h^{\hat{\omega}_i}, g^{\omega_i (\mathbf{y}')^\mathsf{T} \mathbf{v}} h^{\hat{\omega}_i + 1})$ Then, for each $j \in S_i$ sample $\hat{y}'_j \xleftarrow{\mathrm{R}} \mathbb{Z}_2$ subject to the constraint that $\sum_{j \in S_i} \alpha_j \hat{y}'_j = \hat{\omega}_i$.

– For all of the remaining indices $j \in [\ell'] \setminus \bigcup_{i \in I} S_i$, sample $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_2$.

Define $\mathbf{y} \in \mathbb{Z}^{\ell'}_{2p'q'}$ to be the vector where $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod 2$.

4. Next, the simulator computes $\widetilde{\sigma} = (\mathsf{s}, g^{\mathbf{y}^\intercal \mathbf{v}})$ and $\overline{t}_j = g^{s_j \mathbf{y}^\intercal \mathbf{v}} h^{\mathbf{y}^\intercal \mathbf{e}_j}$ for each $j \in S_i$. Next, for each $j \in S_i$ and $\tau \in [T]$, compute $\overline{u}_{\tau,j} \leftarrow \prod_{k \in [\ell']} \overline{z}^{y_k}_{\tau,j,k}$, and set $u_j \leftarrow H(\overline{u}_{1,j}, \ldots, \overline{u}_{T,j})$. It sets $\widetilde{\pi}_i = \{(j, \overline{t}_j, u_j)\}_{j \in S_i}$.
5. Output $\widetilde{\sigma}$ and $\{\widetilde{\pi}_i\}_{i \in I}$.

We now use a hybrid argument to show that $\mathsf{ExptHide}[\mathcal{A}, 0]$ and $\mathsf{ExptHide}[\mathcal{A}, 1]$ are statistically indistinguishable:

- Hyb_0: This is the experiment $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 0]$. Namely, the challenger begins by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$, and gives crs to $\mathcal{A}$. The adversary $\mathcal{A}$ replies with a public key pk. For each challenge query, the challenger samples $(\sigma, r, \{\pi_i\}_{i \in [\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs}, \mathsf{pk})$ and gives r to $\mathcal{A}$ before receiving a set $I \subseteq [\rho]$ chosen by $\mathcal{A}$. It then replies with σ and $\{\pi_i\}_{i \in I}$.

- Hyb_1: This experiment is identical to Hyb_0, except that the challenger computes $(\mathsf{st}_{\mathcal{S}}, \widetilde{\mathsf{crs}}) \leftarrow \mathcal{S}_1(1^\lambda, 1^\rho)$ and uses $\widetilde{\mathsf{crs}}$ in place of crs. Everything else proceeds identically to Hyb_0.

  Specifically, in this experiment, the challenger samples (N, p, q), H, $\mathbf{v}$, $s_1, \ldots, s_{\ell'}$ as specified by $\mathcal{S}_1$ and sets $\widetilde{\mathsf{crs}} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\mathbf{e}_1}, \ldots, g^{s_{\ell'} \mathbf{v}} h^{\mathbf{e}_{\ell'}})$. It gives $\widetilde{\mathsf{crs}}$ to $\mathcal{A}$ to receive a public key $\mathsf{pk} = \{\overline{\mathbf{z}}_{\tau,i}\}_{\tau \in [T], i \in [\ell']}$. On a challenge query, the challenger proceeds as follows:

  1. Check that $\overline{\mathbf{z}}_{\tau,i} \in \mathbb{J}^{\ell'}_N$ for all $\tau \in [T]$ and $i \in [\ell']$, and output $\perp$ otherwise.
  2. Sample $\mathsf{s} \xleftarrow{R} \{0,1\}^\kappa$ and compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$. For each $i \in [\rho]$, let $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$.
  3. Sample $\mathbf{y} \xleftarrow{R} \mathbb{Z}^{\ell'}_{\lfloor N/2 \rfloor}$ and compute for each $j \in [\ell']$ and $\tau \in [T]$,

  $$\overline{c} \leftarrow g^{\mathbf{y}^\intercal \mathbf{v}} \quad \text{and} \quad \overline{t}_j \leftarrow g^{s_j \mathbf{y}^\intercal \mathbf{v}} h^{\mathbf{y}^\intercal \mathbf{e}_j} \quad \text{and} \quad \overline{u}_{\tau,j} \leftarrow \prod_{k \in [\ell']} (\overline{z}_{\tau,j,k})^{y_k}.$$

  Then, for each $j \in [\ell']$, compute $u_j \leftarrow H(\overline{u}_{1,j}, \ldots, \overline{u}_{T,j})$.
  4. For each $i \in [\rho]$, compute $\overline{t}'_i \leftarrow \prod_{j \in S_i} \overline{t}_j^{\alpha_j}$ and $r_i \leftarrow \mathsf{LEQ}(\overline{t}'_i, \overline{t}'_i h)$ and $\pi_i \leftarrow \{(j, \overline{t}_j, u_j)\}_{j \in S_i}$.
  5. The challenger gives r to $\mathcal{A}$ and receives a set $I \subseteq [\rho]$.
  6. The challenger replies with $\sigma = (\mathsf{s}, \overline{c})$ and the set $\{\pi_i\}_{i \in I}$.

- Hyb_2: Same as Hyb_1, except when responding to challenge queries, the challenger samples $\mathbf{y} \xleftarrow{R} \mathbb{Z}^{\ell'}_{2p'q'}$ instead of $\mathbf{y} \xleftarrow{R} \mathbb{Z}^{\ell'}_{\lfloor N/2 \rfloor}$.

- Hyb_3: Same as Hyb_2, except when responding to the challenge queries, the challenger first samples $\mathbf{y}' \xleftarrow{R} \mathbb{Z}^{\ell'}_{p'q'}$ and $\hat{\mathbf{y}}' \xleftarrow{R} \mathbb{Z}^{\ell'}_2$. It defines $\mathbf{y} \in \mathbb{Z}^{\ell'}_{2p'q'}$ to be the vector where $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod 2$.

- Hyb_4: Same as Hyb_3, except when responding to challenge queries, the challenger samples s and $\mathbf{y}'$ as in Hyb_3. Next, for each $i \in [\rho]$, it samples $\hat{\omega}_i \xleftarrow{R} \mathbb{Z}_2$, and for $j \in S_i$, it samples $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_2$ subject to the constraint that $\sum_{j \in S_i} \alpha_j \hat{y}'_j = \hat{\omega}_i$. For all of the remaining indices $j \in [\ell'] \setminus \bigcup_{i \in I} S_i$, sample $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_2$.

- Hyb_5: Same as Hyb_4, except when responding to the challenge queries, the challenger first samples $r \xleftarrow{R} \{0,1\}^\rho$. Then, after sampling s and $\mathbf{y}'$, it computes for each $i \in [\rho]$, $\omega_i \leftarrow \sum_{j \in S_i} \alpha_j s_j \pmod{p'q'}$. Then, it sets $\hat{\omega}_i \in \mathbb{Z}_2$ to be the unique value where $r_i = \mathsf{LEQ}(g^{\omega_i (\mathbf{y}')^\mathsf{T} \mathbf{v}} h^{\hat{\omega}_i}, g^{\omega_i (\mathbf{y}')^\mathsf{T} \mathbf{v}} h^{\hat{\omega}_i + 1})$.

- Hyb_6: Same as Hyb_5, except when responding to the challenge queries, the challenger samples $\hat{\mathbf{y}}' \in \mathbb{Z}_2^{\ell'}$ *after* it receives the challenge set. On each query, after the challenger receives the set $I \subseteq [\rho]$, for each $i \in I$, it samples $\hat{\mathbf{y}}'$ as follows:

  - For each $i \in I$, let $\omega_i \leftarrow \sum_{j \in S_i} \alpha_j s_j \pmod{p'q'}$. Then, set $\hat{\omega}_i \in \mathbb{Z}_2$ to be the unique value where $r_i = \mathsf{LEQ}(g^{\omega_i (\mathbf{y}')^\mathsf{T} \mathbf{v}} h^{\hat{\omega}_i}, g^{\omega_i (\mathbf{y}')^\mathsf{T} \mathbf{v}} h^{\hat{\omega}_i + 1})$ Finally, sample $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_2$ for each $j \in S_i$ subject to the constraint that $\sum_{j \in S_i} \alpha_j \hat{y}'_j = \hat{\omega}_i$.
  - For all of the remaining indices $j \in [\ell'] \setminus \bigcup_{i \in I} S_i$, sample $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_2$.

  The remaining components are constructed as in Hyb_5. This is exactly the distribution in $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 1]$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output of $\mathsf{Hyb}_i(\mathcal{A})$ with adversary $\mathcal{A}$. In the following, we show that the output distribution on each pair of adjacent experiments is statistically indistinguishable (or identically distributed). Because $\mathbb{H} = \{\pm 1\}$ is small, our analysis will rely on a stronger version of Claim C.11:

Claim D.9. *Suppose G is a secure PRG and take $\mathsf{s} \xleftarrow{R} \{0,1\}^\kappa$. Let $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$. Then, with overwhelming probability over the choice of s, the following properties hold:*

- *For any $i \in [\rho]$, $\Pr[\exists j \in \hat{S}_i : \alpha_j \neq 0] = 1 - \mathsf{negl}(\lambda)$.*

- *Let $\mathbf{A} \in \mathbb{Z}_2^{\ell \times \ell' T}$ be any fixed matrix and let $I \subseteq [\rho]$ be any fixed set of indices. For each $i \in I$, let $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$ be the shifted set of indices, and define*

$$J = \{j \in S_i \text{ for some } i \in I \mid j \in [\ell']\} \subseteq [\ell'].$$

  Write $\mathbf{A} = [\ \mathbf{A}_1 \mid \cdots \mid \mathbf{A}_{\ell'}\ ]$, where each $\mathbf{A}_i \in \mathbb{Z}_2^{\ell \times T}$. Let $\hat{\mathbf{A}} \in \mathbb{Z}_2^{\ell \times \lambda |I| T}$ be the submatrix of $\mathbf{A}$ formed by taking only the blocks of $\mathbf{A}$ indexed by the set J (namely, the blocks $\mathbf{A}_j$ for $j \in J$). Next, for $i \in [\rho]$, define $\boldsymbol{\alpha}^{(i)} \in \mathbb{Z}_2^\ell$ as follows:

$$\alpha_j^{(i)} = \begin{cases} \alpha_{j + \ell \cdot (i-1)} & \text{if } j \in \hat{S}_i \\ 0 & \text{otherwise.} \end{cases}$$

  By construction, $\boldsymbol{\alpha}^{(i)}$ contains exactly λ non-zero entries (as specified by the set $\hat{S}_i$). Then, for any $i \in [\rho] \setminus I$,

$$\Pr[\boldsymbol{\alpha}^{(i)} \in \mathrm{span}(\hat{\mathbf{A}})] = \Pr[\exists \mathbf{v} \in \mathbb{Z}_2^{\lambda |I| T} : \boldsymbol{\alpha}^{(i)} = \hat{\mathbf{A}} \mathbf{v}] = \mathsf{negl}(\lambda).$$

Proof. As in the proof of Claim C.11, it suffices to check that these properties hold for truly random $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \xleftarrow{R} \mathcal{T}_{\lambda,\ell}^\rho \times \mathbb{Z}_2^{\rho \ell}$. We show each property below:

- Since $\boldsymbol{\alpha} \xleftarrow{R} \mathbb{Z}_2^{\rho\ell}$, for all $j \in [\rho\ell]$, $\Pr[\alpha_j = 0] = 1/2$. Since $|\hat{S}_i| = \lambda$, we have that $\Pr[\forall j \in \hat{S}_i : \alpha_j = 0] = 1/2^\lambda = \mathsf{negl}(\lambda)$, and the claim holds.

- Let $\hat{\mathbf{A}}$ be the matrix defined above and let $\hat{\mathbf{a}}_j^\mathsf{T} \in \mathbb{Z}_2^{\lambda|I|T}$ denote the j^{th} row of $\hat{\mathbf{A}}$. Let $n = \mathrm{rank}(\hat{\mathbf{A}}) \le \lambda|I|T \le \lambda\rho T = \ell/(36\lambda)$. This means that there exists a collection of indices $j_1, \ldots, j_n \in [\ell]$ such that the collection $\{\hat{\mathbf{a}}_{j_1}^\mathsf{T}, \ldots, \hat{\mathbf{a}}_{j_n}^\mathsf{T}\}$ is linearly independent and $\mathrm{span}(\hat{\mathbf{A}}^\mathsf{T}) = \mathrm{span}(\{\hat{\mathbf{a}}_{j_1}^\mathsf{T}, \ldots, \hat{\mathbf{a}}_{j_n}^\mathsf{T}\})$. By construction $\hat{\mathbf{A}}$ only depends on $\mathbf{A}$ and the sets $\hat{S}_i$ where $i \in I$. This means that we can sample the sets $\hat{S}_i$ where $i \notin I$ *after* fixing $\hat{\mathbf{A}}$ and the set of indices $\{j_1, \ldots, j_n\}$. In this case, we will show that if we sample $\hat{S}_i \xleftarrow{R} \mathcal{T}_{\lambda,\ell}$, then with negligible probability, $|\hat{S}_i \cap \{j_1, \ldots, j_n\}| \ge \lambda/2$. Let $K = |\hat{S}_i \cap \{j_1, \ldots, j_n\}|$. In the proof of Claim C.11, it was sufficient to show that $K < \lambda$ with overwhelming probability because we were working over a super-polynomial-size field. Here, we require a stronger property that $K < \lambda/2$ with overwhelming probability because we are working over the binary field. Take any $\lambda \ge k \ge \lambda/2$.

$$\Pr[K = k] = \frac{\binom{n}{k}\binom{\ell-n}{\lambda-k}}{\binom{\ell}{\lambda}} \le \left(\frac{en}{k}\right)^k (e\ell)^{\lambda-k} \left(\frac{\lambda}{\ell}\right)^\lambda,$$

where we have used the fact that $\left(\frac{n}{k}\right)^k \le \binom{n}{k} \le \left(\frac{en}{k}\right)^k$. Since $n \le \ell/(36\lambda)$ and $\lambda \ge k \ge \lambda/2$, we have that

$$\Pr[K = k] \le \frac{(e\lambda)^\lambda}{\ell^k}\left(\frac{n}{k}\right)^k \le \frac{(e\lambda)^\lambda}{\ell^k}\left(\frac{2\ell}{36\lambda^2}\right)^k = \frac{(e\lambda)^\lambda 2^k}{(36\lambda^2)^k} \le \frac{(2e)^\lambda}{36^{\lambda/2}} = \frac{(2e)^\lambda}{6^\lambda} = \mathsf{negl}(\lambda).$$

Then, by a union bound $\Pr[\lambda \ge K \ge \lambda/2] = \mathsf{negl}(\lambda)$. Thus, with overwhelming probability, $K = |\hat{S}_i \cap \{j_1, \ldots, j_n\}| < \lambda/2$. This means that there exist at least $\lambda/2$ indices $j_k^* \in \hat{S}_i$ where $j_k^* \notin \{j_1, \ldots, j_n\}$. Now as in the proof of Claim C.11, since $\hat{\mathbf{a}}_{j_k^*} \in \mathrm{span}(\hat{\mathbf{A}}^\mathsf{T})$, there exists scalars $\beta_1, \ldots, \beta_n \in \mathbb{Z}_2$ such that $\hat{\mathbf{a}}_{j_k^*}^\mathsf{T} = \sum_{\gamma\in[n]} \beta_\gamma \hat{\mathbf{a}}_{j_\gamma}^\mathsf{T}$. This means that if there exists $\mathbf{v} \in \mathbb{Z}_2^{\lambda|I|T}$ such that $\boldsymbol{\alpha}^{(i)} = \hat{\mathbf{A}}\mathbf{v}$, then

$$\alpha_{j_k^*}^{(i)} = \hat{\mathbf{a}}_{j_k^*}^\mathsf{T}\mathbf{v} = \sum_{\gamma\in[n]} \beta_\gamma \hat{\mathbf{a}}_{j_\gamma}^\mathsf{T}\mathbf{v} = \sum_{\gamma\in[n]} \beta_\gamma \alpha_{j_\gamma}^{(i)}. \tag{D.5}$$

Since $j_k^* \in \hat{S}_i$, by construction, $\alpha_{j_k^*}^{(i)} = \alpha_{j_k^*+\ell\cdot(i-1)}$, which is uniform over $\mathbb{Z}_2$ and independent of β_γ and $\alpha_{j_\gamma}^{(i)}$ for all $\gamma \in [n]$. Thus, over the randomness of $\boldsymbol{\alpha}$, Eq. (D.5) holds with probability $1/2$ for each index j_k^*. Thus, for any $i \in [\rho] \setminus I$, with overwhelming probability over the choice of $\hat{S}_i$, there are at least $\lambda/2$ such indices $j_k^* \in \hat{S}_i$, and since each $\alpha_{j_k^*}^{(i)}$ is sampled independently and uniformly over $\mathbb{Z}_2$, we have that

$$\Pr[\exists \mathbf{v} \in \mathbb{Z}_2^{\lambda|I|T} : \boldsymbol{\alpha}^{(i)} = \hat{\mathbf{A}}\mathbf{v}] \le \frac{1}{2^{\lambda/2}} = \mathsf{negl}(\lambda). \qquad \square$$

Lemma D.10. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \equiv \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. Since $\mathcal{S}_1(1^\lambda, 1^\rho)$ samples $\widetilde{\mathsf{crs}}$ using the same procedure as $\mathsf{Setup}(1^\lambda, 1^\rho)$, the output distributions of Hyb_0 and Hyb_1 are identical. $\square$

Lemma D.11. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. Follows by the same argument as the proof of Lemma D.5. □

Lemma D.12. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_2(\mathcal{A}) \equiv \mathsf{Hyb}_3(\mathcal{A})$.*

Proof. Follows by the Chinese remainder theorem (as in the proof of Lemma D.6). □

Lemma D.13. *If G is a secure PRG, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_3(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_4(\mathcal{A})$.*

Proof. In Hyb_3, the challenger samples $\hat{\mathbf{y}}' \xleftarrow{R} \mathbb{Z}_2^{\ell'}$ while in Hyb_4, the challenger samples $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_2$ subject to the constraint that $\sum_{j \in S_i} \alpha_j \hat{y}'_j = \hat{\omega}_i$ where $\hat{\omega}_i \xleftarrow{R} \mathbb{Z}_2$ for $j \in S_i$ and $i \in [\rho]$. For the indices $j \in [\ell'] \setminus \bigcup_{i \in I} S_i$, $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_2$. These distributions are identical as long as there exists some $j \in S_i$ where $\alpha_j \neq 0$ for each $i \in [\rho]$. Since G is a secure PRG, this follows by Claim D.9. □

Lemma D.14. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_4(\mathcal{A}) \equiv \mathsf{Hyb}_5(\mathcal{A})$.*

Proof. First, $\mathbf{y}'$ is sampled identically in the two distributions. It suffices to show that once we fix the PRG seed s and the vector $\mathbf{y}' \in \mathbb{Z}_{p'q'}^{\ell'}$, there is a one-to-one correspondence between the value of $\hat{\omega}_i$ and the value of r_i. In Hyb_4, $r_i = \mathsf{LEQ}(\overline{t}'_i, \overline{t}'_i h)$ where

$$\overline{t}'_i = \prod_{j \in S_i} \overline{t}_j^{\alpha_j} = \prod_{j \in S_i} (g^{s_j \mathbf{y}^\mathsf{T} \mathbf{v}} h^{\mathbf{y}^\mathsf{T} \mathbf{e}_j})^{\alpha_j}.$$

Since g generates a subgroup of order $p'q'$, and $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$, we have that in Hyb_3, $g^{\alpha_j s_j \mathbf{y}^\mathsf{T} \mathbf{v}} = g^{\alpha_j s_j (\mathbf{y}')^\mathsf{T} \mathbf{v}}$. Similarly, since h has order 2, $h^{\mathbf{y}^\mathsf{T} \mathbf{e}_j} = h^{y_j} = h^{\hat{y}'_j}$. This means that

$$\overline{t}'_i = \prod_{j \in S_i} (g^{s_j \mathbf{y}^\mathsf{T} \mathbf{v}} h^{\mathbf{y}^\mathsf{T} \mathbf{e}_j})^{\alpha_j} = \prod_{j \in S_i} g^{\alpha_j s_j (\mathbf{y}')^\mathsf{T} \mathbf{v}} h^{\alpha_j \hat{y}'_j} = g^{\omega_i (\mathbf{y}')^\mathsf{T} \mathbf{v}} h^{\hat{\omega}_i},$$

where $\omega_i = \sum_{j \in S_i} \alpha_j s_j$ and $\hat{\omega}_i = \sum_{j \in S_i} \alpha_j \hat{y}'_j$. Since $r_i = \mathsf{LEQ}(\overline{t}'_i, \overline{t}'_i h)$, we conclude that once we fix the seed s (which together with the CRS determines the value of ω_i) and the vector $\hat{\mathbf{y}}'$, then the value of r_i is entirely dependent on the value of $\hat{\omega}_i \pmod 2$. By the same argument as in the proof of Lemma D.7, there is a one-to-one correspondence between the value of $\hat{\omega}_i$ and r_i, and both distributions enforce this correspondence. As such, these two distributions are identical. □

Lemma D.15. *If G is a secure PRG, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_5(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_6(\mathcal{A})$.*

Proof. The challenger samples $\widetilde{\mathsf{crs}}$ identically in the two experiments, so N, g, h, $\mathbf{v}$, $s_1, \ldots, s_{\ell'}$ are identically distributed in the two experiments. so it suffices to consider its responses to the challenge queries. In both experiments, on each query, the challenger starts by sending the adversary a random string $r \xleftarrow{R} \{0,1\}^\rho$, and the adversary replies with a set $I \subseteq [\rho]$. The challenger then replies with a commitment $\sigma = (\mathsf{s}, \overline{c})$ and a set of proofs $\{\pi_i\}_{i \in I}$, where $\pi_i = \{(j, \overline{t}_j, u_j)\}_{j \in S_i}$. We show that the challenger's responses are statistically indistinguishable in the two experiments. To this end, we define the following variables:

- For $i \in [\rho]$, let $\hat{\mathbf{y}}^{(i)} \in \mathbb{Z}_2^\ell$ be the vector where $\hat{y}_j^{(i)} = \hat{y}'_{j+\ell \cdot (i-1)} \in \mathbb{Z}_N$. In other words, $(\hat{\mathbf{y}}')^\mathsf{T} = [\ (\hat{\mathbf{y}}^{(1)})^\mathsf{T} \mid \cdots \mid (\hat{\mathbf{y}}^{(\rho)})^\mathsf{T}\ ] \in \mathbb{Z}_2^{\ell'}$.

- For $i \in [\rho]$, define $\boldsymbol{\alpha}^{(i)} = (\alpha_1^{(i)}, \ldots, \alpha_\ell^{(i)})^\top \in \mathbb{Z}_N^\ell$ as follows:

$$\alpha_j^{(i)} = \begin{cases} \alpha_{j+\ell\cdot(i-1)} & \text{if } j \in \hat{S}_i \\ 0 & \text{otherwise.} \end{cases}$$

We now consider the different components in the two experiments:

- The string r is independently and uniformly sampled from $\{0,1\}^\rho$ in both experiments.
- The seed s is independently and uniformly sampled from $\{0,1\}^\kappa$ in both experiments, so the seed s, the sets $S_1, \ldots, S_\rho$, and the vector $\boldsymbol{\alpha}$ are identically distributed as well.
- The vector $\mathbf{y}'$ is sampled uniformly from $\mathbb{Z}_{p'q'}^{\ell'}$ in both experiments. Since g generates a group of order $p'q'$, the commitment $\overline{c} = g^{\mathbf{y}^\mathsf{T}\mathbf{v}} = g^{(\mathbf{y}')^\mathsf{T}\mathbf{v}}$ is identically distributed in Hyb_5 and Hyb_6.
- For all $i \in I$, both experiments set $\omega_i \leftarrow \sum_{j \in S_i} \alpha_j s_j \pmod{p'q'}$ and sample $\hat{\omega}_i \xleftarrow{\mathrm{R}} \mathbb{Z}_2$ subject to $r_i = \mathsf{LEQ}(g^{\omega_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{\omega}_i}, g^{\omega_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{\omega}_i+1})$. This means that the vectors $\hat{\mathbf{y}}^{(i)}$ for $i \in I$ are identically distributed in the two experiments. Then, for $j \in S_i$, we have that $\overline{t}_j = g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_j}$, so the components $\overline{t}_j$ for $j \in S_i$ and $i \in I$ are identically distributed.

It suffices to show that the remaining components $\{u_j\}_{j \in S_i}$ for $i \in I$ are drawn from statistically indistinguishable distributions (from the view of the adversary). Since $u_j = H(\overline{u}_{1,j}, \ldots, \overline{u}_{T,j})$, it suffices to show that the elements $\{\overline{u}_{\tau,j}\}_{\tau \in [T], j \in S_i}$ are statistically indistinguishable in the two experiments. First, let

$$J = \{j \in S_i \text{ for some } i \in I \mid j \in [\ell']\}$$

be the set of indices that appear in some set S_i for $i \in I$. Next, let $\overline{\mathbf{z}}_{\tau,j} = g^{\mathbf{z}_{\tau,j}} h^{\hat{\mathbf{z}}_{\tau,j}}$ be the subgroup decomposition of the public-key components. Then, in both experiments, the challenger computes $\overline{u}_{\tau,j}$ as

$$\overline{u}_{\tau,j} = \prod_{k \in [\ell']} \overline{z}_{\tau,j,k}^{y_k} = g^{(\mathbf{y}')^\mathsf{T}\mathbf{z}_{\tau,j}} h^{(\hat{\mathbf{y}}')^\mathsf{T}\hat{\mathbf{z}}_{\tau,j}}.$$

Since $\mathbf{y}'$ is identically distributed in Hyb_5 and Hyb_6, it suffices to consider the distribution of $\overline{u}_{\tau,j}$ in the $\mathbb{H}$ subgroup, or equivalently, the distribution of $(\hat{\mathbf{y}}')^\mathsf{T}\hat{\mathbf{z}}_{\tau,j}$ over $\mathbb{Z}_2$. Let $\hat{\mathbf{Z}}_i \in \mathbb{Z}_2^{\ell' \times T}$ be the matrix whose columns are $\hat{\mathbf{z}}_{1,i}, \ldots, \hat{\mathbf{z}}_{T,i}$, and let $\hat{\mathbf{Z}} \in \mathbb{Z}_2^{\ell' \times \ell' T}$ be the matrix

$$\hat{\mathbf{Z}} = [\ \hat{\mathbf{Z}}_1 \mid \cdots \mid \hat{\mathbf{Z}}_{\ell'}\ ].$$

Let $\hat{\mathbf{Z}}' \in \mathbb{Z}_2^{\ell' \times \lambda |I| T}$ be the submatrix of $\hat{\mathbf{Z}}$ formed by taking only the blocks of $\hat{\mathbf{Z}}$ indexed by the set J (namely, the blocks $\hat{\mathbf{Z}}_j$ for $j \in J$). In particular, the values of $h^{(\hat{\mathbf{y}}')^\mathsf{T}\hat{\mathbf{Z}}'} \in \mathbb{H}^{\lambda |I| T}$ precisely coincide with the components in the $\mathbb{H}$-subgroup of $\overline{u}_{\tau,j}$ for all $\tau \in [T]$, $j \in S_i$ and $i \in I$. Thus, it suffices to show that the distributions of $(\hat{\mathbf{y}}')^\mathsf{T}\hat{\mathbf{Z}}' \in \mathbb{Z}_2^{\lambda |I| T}$ are statistically indistinguishable in the two experiments. First, for $i \in [\rho]$, let $\hat{\mathbf{Z}}^{(i)} \in \mathbb{Z}_2^{\ell \times \lambda |I| T}$ be matrices such that

$$\hat{\mathbf{Z}}' = \begin{bmatrix} \hat{\mathbf{Z}}^{(1)} \\ \vdots \\ \hat{\mathbf{Z}}^{(\rho)} \end{bmatrix},$$

This means that

$$(\hat{\mathbf{y}}')^{\mathsf{T}}\hat{\mathbf{Z}}' = \sum_{i\in[\rho]} \left((\hat{\mathbf{y}}^{(i)})^{\mathsf{T}}\hat{\mathbf{Z}}^{(i)}\right) \in \mathbb{Z}_2^{\lambda|I|T}. \tag{D.6}$$

By definition, $\hat{\mathbf{Z}}$ is a *fixed* matrix (determined by the CRS and the adversary's public key) and independent of $\hat{\mathbf{y}}'$. Since each $\hat{\mathbf{y}}^{(i)}$ is sampled independently, we can consider each term individually in this summation:

- If $i \in I$, then as argued above, $\hat{\mathbf{y}}^{(i)}$ is identically distributed in Hyb_5 and Hyb_6, and correspondingly, so is the product $(\hat{\mathbf{y}}^{(i)})^{\mathsf{T}}\hat{\mathbf{Z}}^{(i)}$.
- If $i \notin I$, then the $\hat{\mathbf{y}}^{(i)}$ in the two experiments are sampled from distinct distributions. In Hyb_5, $\hat{\mathbf{y}}^{(i)}$ is uniform over $\mathbb{Z}_2^\ell$ subject to $\hat{\omega}_i = \sum_{j\in[\ell]} \alpha_j^{(i)}\hat{y}_j^{(i)}$, while in Hyb_6, $\hat{\mathbf{y}}^{(i)}$ is uniform over $\mathbb{Z}_2^\ell$.

  Since $i \notin I$ and $\hat{\mathbf{Z}}$ is a fixed matrix, we can appeal to Claim D.9 (for the i^{th} block of $\hat{\mathbf{Z}}$) and conclude that with overwhelming probability, $\boldsymbol{\alpha}^{(i)} \notin \mathrm{span}(\hat{\mathbf{Z}}^{(i)})$. By Claim C.12, this means that the distribution of $(\hat{\mathbf{y}}^{(i)})^{\mathsf{T}}\hat{\mathbf{Z}}^{(i)}$ in Hyb_5 (where $\hat{\mathbf{y}}^{(i)}$ is uniform subject to a linear constraint $\boldsymbol{\alpha}^{(i)} \notin \mathrm{span}(\hat{\mathbf{Z}}^{(i)})$) is statistically indistinguishable from its distribution in Hyb_6 (where $\hat{\mathbf{y}}^{(i)}$ is uniform).

Since every term in Eq. (D.6) is either statistically indistinguishable or identically distributed in the two experiments, we conclude that $(\hat{\mathbf{y}}')^{\mathsf{T}}\hat{\mathbf{Z}}'$ is also statistically indistinguishable in the two experiments. Correspondingly, this means that the components $\overline{u}_{\tau,j}$ for $\tau \in [T]$, $j \in S_i$, and $i \in I$ are also statistically indistinguishable in the two experiments. □

Since each consecutive pair of hybrid experiments is statistically indistinguishable (or identically distributed), the theorem follows. □

E Dual-Mode Hidden-Bits Generators from DCR

In this section, we show how to construct dual-mode hidden-bits generators from the DCR assumption. As in Section 5, we begin with a basic construction (Construction E.2) and then show how to extend it to obtain one with security against malicious verifiers (Construction E.16). Both constructions are structurally similar to the corresponding constructions from QR (Constructions 5.3 and 5.9, respectively). One difference between the two construction is that we now use a universal hash function to extract the hidden bit rather than the LEQ predicate. In addition, we rely on a slightly different verification check. This is because in the QR constructions, the public key, commitment, and proofs are all elements in $\mathbb{J}_N$ and membership in $\mathbb{J}_N$ is efficiently-decidable. The analog of $\mathbb{J}_N$ in the DCR setting is the subgroup of quadratic residues in $\mathbb{Z}^*_{N^2}$, which is not an efficiently-decidable language. Thus, a malicious prover or verifier could publish public-keys, commitments, or proofs that are quadratic non-residues over $\mathbb{Z}^*_{N^2}$ to try and break soundness or zero-knowledge. We handle this case by adjusting the verification relation.

E.1 Dual-Mode Hidden-Bits Generator from DCR

In this section, we construct a standard dual-mode hidden-bits generator from the DCR assumption (without malicious security). This is an analog of Construction 5.3 from the QR assumption. Our

construction is inspired by the trapdoor hash function of [DGI+19, Appendix A]. We begin by recalling some basic notation as well as the DCR assumption.

Notation. Let $N = pq$ be a product of safe primes p, q. Our construction works over the group $\mathbb{Z}^*_{N^2}$, which splits into two subgroups $\mathbb{H} := \{(1+N)^i \mid i \in [N]\}$ and $\mathbb{NR}_N := \{x^N \mid x \in \mathbb{Z}^*_{N^2}\}$ of coprime orders N and $\varphi(N)$, respectively. In our analysis, we will use the fact that if $x \in \mathbb{Z}^*_{N^2} = \mathbb{NR}_N \times \mathbb{H}$, then $x^2 \in \mathbb{NR}_{2N} \times \mathbb{H}$. When N is a product of safe primes $p = 2p' + 1$, $q = 2q' + 1$, the subgroup $\mathbb{NR}_{2N}$ of $2N^{\text{th}}$ residues is cyclic (and has order $p'q'$). In the following, we write g to denote a generator of $\mathbb{NR}_{2N}$ and $h = (1+N)$ to denote a generator of $\mathbb{H}$. We will typically denote elements of $\mathbb{NR}_{2N} \times \mathbb{H}$ with a bar (e.g., $\bar{c}$, $\bar{t}$). In the analysis, we often analyze elements of $\mathbb{NR}_{2N} \times \mathbb{H}$ by examining their components in their respective subgroups: we write $\bar{c} = g^c h^{\hat{c}} \in \mathbb{NR}_{2N} \times \mathbb{H}$ where $c \in \mathbb{Z}_{p'q'}$ and $\hat{c} \in \mathbb{Z}_N$.

Definition E.1 (Decisional Composite Residuosity Assumption [Pai99])**.** A safe prime modulus sampler $\mathsf{SampleModulus}$ satisfies the decisional composite residuosity (DCR) assumption if for all efficient adversaries $\mathcal{A}$, and sampling $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$, $x \xleftarrow{R} \mathbb{Z}^*_{N^2}$,

$$\left|\Pr[\mathcal{A}(N, x) = 1] - \Pr[\mathcal{A}(N, x^N)]\right| = \mathsf{negl}(\lambda).$$

Construction E.2 (Dual-Mode Hidden-Bits Generator from DCR)**.** Our DCR-based dual-mode hidden-bits generator (HBG) goes as follows:

- $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode}) \to (\mathsf{crs}, \mathsf{sk})$: Sample $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$. Let g be a generator of $\mathbb{NR}_{2N}$ and $h = 1 + N$ be the generator of $\mathbb{H}$. The setup algorithm samples a vector $\mathbf{v} \xleftarrow{R} \mathbb{Z}^\rho_{\lfloor N^2/4 \rfloor}$, scalars $s_1, \ldots, s_\rho \xleftarrow{R} \mathbb{Z}_{\lfloor N^2/4 \rfloor}$ and sets $\hat{\mathbf{w}}_i \in \mathbb{Z}^\rho_N$ for $i \in [\rho]$ as follows:
  - If $\mathsf{mode} = \mathsf{hiding}$, set $\hat{\mathbf{w}}_i \leftarrow \mathbf{e}_i$, where $\mathbf{e}_i \in \mathbb{Z}^\rho_N$ is the i^{th} basis vector.
  - If $\mathsf{mode} = \mathsf{binding}$, set $\hat{\mathbf{w}}_i \leftarrow \mathbf{0}$.

  Finally, it sample a hash function $H \xleftarrow{R} \mathcal{H}$ where $\mathcal{H}$ is a family of hash functions with domain $\mathbb{Z}_{N^2}$ and range $\{0,1\}$. Output $\mathsf{crs} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\hat{\mathbf{w}}_1}, \ldots, g^{s_\rho \mathbf{v}} h^{\hat{\mathbf{w}}_\rho})$.
- $\mathsf{KeyGen}(\mathsf{crs}) \to (\mathsf{pk}, \mathsf{sk})$: On input $\mathsf{crs} = (N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_\rho)$, sample $a, b_1, \ldots, b_\rho \xleftarrow{R} \mathbb{Z}_{\lfloor N^2/4 \rfloor}$. Output $\mathsf{pk} = (\overline{\mathbf{v}}^{b_1} \overline{\mathbf{w}}_1^a, \ldots, \overline{\mathbf{v}}^{b_\rho} \overline{\mathbf{w}}_\rho^a)$ and $\mathsf{sk} = (a, b_1, \ldots, b_\rho)$.
- $\mathsf{GenBits}(\mathsf{crs}, \mathsf{pk}) \to (\sigma, r, \{\pi_i\}_{i \in [\rho]})$: On input $\mathsf{crs} = (N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_\rho)$ and $\mathsf{pk} = (\overline{\mathbf{z}}_1, \ldots, \overline{\mathbf{z}}_\rho)$, sample $\mathbf{y} \xleftarrow{R} \mathbb{Z}^\rho_{\lfloor N^2/4 \rfloor}$ and compute for all $i \in [\rho]$:

$$\bar{c} \leftarrow \prod_{j \in [\rho]} \overline{v}_j^{y_j} \quad \text{and} \quad \bar{t}_i \leftarrow \prod_{j \in [\rho]} \overline{w}_{i,j}^{y_j} \quad \text{and} \quad \bar{u}_i \leftarrow \prod_{j \in [\rho]} (\overline{z}_{i,j}^2)^{y_j}.$$

  For each $i \in [\rho]$, let $r_i \leftarrow H(\bar{t}_i^2) \in \{0,1\}$. Output $\sigma = \bar{c}$, r, and $\pi = \{(\bar{t}_i, \bar{u}_i)\}_{i \in [\rho]}$.
- $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$: On input $\mathsf{crs} = (N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_\rho)$, $\mathsf{sk} = (a, b_1, \ldots, b_\rho)$, $\sigma = \bar{c}$, $i \in [\rho]$, $r_i \in \{0,1\}$, and $\pi_i = (\bar{t}_i, \bar{u}_i)$, output 1 if $\bar{u}_i = (\bar{t}_i^a \bar{c}^{b_i})^2$ and $r_i = H(\bar{t}_i^2)$. Otherwise, output 0.

Correctness and security analysis. We now state the correctness and security theorems for Construction E.2 and give the proofs in Appendix E.1.1.

Theorem E.3 (Correctness)**.** *Construction E.2 is correct.*

Theorem E.4 (Succinctness)**.** *Construction E.2 is succinct.*

Theorem E.5 (CRS Indistinguishability)**.** *Suppose the* DCR *assumption holds with respect to* SampleModulus*. Then, Construction E.2 satisfies CRS indistinguishability.*

Theorem E.6 (Statistical Binding in Binding Mode)**.** *Construction E.2 satisfies statistical binding in binding mode.*

Theorem E.7 (Statistical Simulation in Hiding Mode)**.** *If $\mathcal{H}$ is a universal hash, then Construction E.2 satisfies statistical simulation in hiding mode.*

E.1.1 Analysis of Construction E.2 (Dual-Mode HBG from DCR)

In this section, we give the proofs for the correctness and security theorems (Theorems E.3 to E.7) for the dual-mode hidden-bits generator from the DCR assumption (Construction E.2).

Proof of Theorem E.3 (Correctness). Fix $\lambda \in \mathbb{N}$, a polynomial $\rho = \rho(\lambda)$, an index $i \in [\rho]$, and a mode $\mathsf{mode} \in \{\mathsf{binding}, \mathsf{hiding}\}$. Let $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode})$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$, and $(\sigma, r, \{\pi_i\}_{i\in[\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs}, \mathsf{pk})$. By construction, $\mathsf{crs} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1\mathbf{v}}h^{\hat{\mathbf{w}}_1}, \ldots, g^{s_\rho\mathbf{v}}h^{\hat{\mathbf{w}}_\rho})$ for some $\mathbf{v} \in \mathbb{Z}^\rho_{\lfloor N^2/4 \rfloor}$ and $\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_\rho \in \mathbb{Z}^\rho_N$. Then, $\mathsf{sk} = (a, b_1, \ldots, b_\rho)$ and

$$\mathsf{pk} = \left(g^{b_1\mathbf{v}+as_1\mathbf{v}}h^{a\hat{\mathbf{w}}_1}, \ldots, g^{b_\rho\mathbf{v}+as_\rho\mathbf{v}}h^{a\hat{\mathbf{w}}_\rho}\right) = \left(g^{(as_1+b_1)\mathbf{v}}h^{a\hat{\mathbf{w}}_1}, \ldots, g^{(as_\rho+b_\rho)\mathbf{v}}h^{a\hat{\mathbf{w}}_\rho}.\right)$$

Similarly, we have $\sigma = \overline{c} = \prod_{j\in[\rho]} g^{v_j y_j} = g^{\mathbf{y}^\mathsf{T}\mathbf{v}}$, $r_i = H(\overline{t}_i^2)$ and $\pi_i = (\overline{t}_i, \overline{u}_i)$ where

$$\overline{t}_i = \prod_{j\in[\rho]} \left(g^{s_i v_j} h^{\hat{w}_{i,j}}\right)^{y_j} = g^{s_i\mathbf{y}^\mathsf{T}\mathbf{v}}h^{\mathbf{y}^\mathsf{T}\hat{\mathbf{w}}_i}$$
$$\overline{u}_i = \prod_{j\in[\rho]} \left(g^{(as_i+b_i)v_j} h^{a\hat{w}_{i,j}}\right)^{2y_j} = (g^{(as_i+b_i)\mathbf{y}^\mathsf{T}\mathbf{v}}h^{a\mathbf{y}^\mathsf{T}\hat{\mathbf{w}}_i})^2.$$

Consider now the behavior of $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$. By construction, $r_i = H(\overline{t}_i^2)$, so it suffices to check that $\overline{u}_i = (\overline{t}_i^a \overline{c}^{b_i})^2$. From the above relations,

$$(\overline{t}_i^a \overline{c}^{b_i})^2 = \left(g^{s_i\mathbf{y}^\mathsf{T}\mathbf{v}}h^{\mathbf{y}^\mathsf{T}\hat{\mathbf{w}}_i}\right)^{2a}\left(g^{\mathbf{y}^\mathsf{T}\mathbf{v}}\right)^{2b_i} = \left(g^{(as_i+b_i)\mathbf{y}^\mathsf{T}\mathbf{v}}h^{a\mathbf{y}^\mathsf{T}\hat{\mathbf{w}}_i}\right)^2 = \overline{u}_i,$$

and the verification algorithm outputs 1. □

Proof of Theorem E.4 (Succinctness). The size of the commitment in Construction E.2 is a single element in $\mathbb{Z}_{N^2}$, which has length $2\lceil \log N \rceil = \mathsf{poly}(\lambda)$. □

Proof of Theorem E.5 (CRS Indistinguishability). First, under the DCR assumption, for $N \leftarrow \mathsf{SampleModulus}(1^\lambda)$, the following two distributions are computationally indistinguishable:

$$\{x \xleftarrow{R} \mathbb{Z}^*_{N^2} : (N, x)\} \overset{c}{\approx} \{x \xleftarrow{R} \mathbb{Z}^*_{N^2} : (N, x^N)\}.$$

This means that the following two distributions are also computationally indistinguishable:

$$\{x \xleftarrow{R} \mathbb{Z}^*_{N^2} : (N, x^2)\} \overset{c}{\approx} \{x \xleftarrow{R} \mathbb{Z}^*_{N^2} : (N, x^{2N})\}.$$

This means that the uniform distribution over $\mathbb{NR}_{2N} \times \mathbb{H}$ is computationally indistinguishable from the uniform distribution over $\mathbb{NR}_{2N}$. We can now appeal to the following theorem from [BG10]:

Claim E.8 ([BG10, §B.2])**.** *Suppose a safe prime product modulus sampler* SampleModulus *satisfies the* DCR *assumption. Then, for all polynomials $\rho = \rho(\lambda)$, all fixed vectors $\hat{\mathbf{w}} \in \{0,1\}^\rho$, and all efficient adversaries $\mathcal{A}$, if we sample $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$, $\mathbf{v} \xleftarrow{R} \mathbb{Z}^\rho_{p'q'}$, $s \xleftarrow{R} \mathbb{Z}_{p'q'}$, where $p = 2p' + 1$ and $q = 2q' + 1$, we have that*

$$\left|\Pr[\mathcal{A}(N, g, g^{\mathbf{v}}, g^{s\mathbf{v}}) = 1] - \Pr[\mathcal{A}(N, g, g^{\mathbf{v}}, g^{s\mathbf{v}} h^{\hat{\mathbf{w}}}) = 1]\right| = \mathsf{negl}(\lambda),$$

where g is a generator of $\mathbb{NR}_{2N}$ and $h = (1 + N)$ is a generator of $\mathbb{H}$.

First, Claim E.8 holds even if $\mathbf{v} \xleftarrow{R} \mathbb{Z}^\rho_{\lfloor N^2/4 \rfloor}$ and $s \xleftarrow{R} \mathbb{Z}_{\lfloor N^2/4 \rfloor}$ since the statistical distance between $\{r \xleftarrow{R} \mathbb{Z}_{\lfloor N^2/4 \rfloor} : r \bmod p'q'\}$ and $\mathsf{Uniform}(\mathbb{Z}_{p'q'})$ is negligible. The theorem now follows by an analogous hybrid argument as in the proof of Theorem 5.6. □

Proof of Theorem E.6 (Statistical Binding). Recall that in binding mode, the common reference string is given by

$$\mathsf{crs} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{w}}, \ldots, g^{s_\rho \mathbf{w}}).$$

We define the (inefficient) Open algorithm as follows:

- $\mathsf{Open}(\mathsf{crs}, \sigma) \to r$: On input a $\mathsf{crs} = (N, g, h, g^{\mathbf{v}}, g^{s_1 \mathbf{v}}, \ldots, g^{s_\rho \mathbf{v}})$ and a commitment $\sigma = \bar{c} \in \mathbb{Z}^*_{N^2}$ (outputting $\perp$ if the components do not have this form), the open algorithm recovers $s_1, \ldots, s_\rho$. It computes $r_i \leftarrow H((\bar{c}^{s_i})^2)$ for each $i \in [\rho]$ and outputs r.

To complete the proof, we use a hybrid argument:

- Hyb_0: This is the real soundness experiment. The challenger samples $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{binding})$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$ and gives $(\mathsf{crs}, \mathsf{pk})$ to $\mathcal{A}$. Here,

  $$\mathsf{crs} = \left(N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}}, \ldots, g^{s_\rho \mathbf{v}}\right) \quad \text{and} \quad \mathsf{pk} = \left(g^{(as_1 + b_1)\mathbf{v}}, \ldots, g^{(as_\rho + b_\rho)\mathbf{v}}\right).$$

  The adversary can make queries to the verification oracle, and on each query (σ, i, r_i, π_i), the challenger replies with $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$. At the end of the game, the adversary outputs a tuple $(\sigma^*, i^*, r^*, \pi^*)$ and the output of the experiment is 1 if $r^* \neq r_i$ where $r \leftarrow \mathsf{Open}(\mathsf{crs}, \sigma^*)$ and $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma^*, i^*, r^*, \pi^*) = 1$.

- Hyb_1: Same as Hyb_0 except the challenger samples the scalars $s_1, \ldots, s_\rho$ and the secret key components $a, b_1, \ldots, b_\rho$ uniformly at random from $\mathbb{Z}_{Np'q'}$ (instead of $\mathbb{Z}_{\lfloor N^2/4 \rfloor}$).

- Hyb_2: Same as Hyb_1 except the challenger computes $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$ using the following modified procedure. Let $\sigma = \overline{c} \in \mathbb{Z}^*_{N^2}$, and write $\overline{c}^2 = g^c h^{\hat{c}}$ for some $c \in \mathbb{Z}_{p'q'}$ and $\hat{c} \in \mathbb{Z}_N$. Similarly, parse $\pi_i = (\overline{t}_i, \overline{u}_i) \in (\mathbb{Z}^*_{N^2})^2$ and write $\overline{t}_i^2$ as $g^{t_i} h^{\hat{t}_i}$ for $t_i \in \mathbb{Z}_{p'q'}$ and $\hat{t}_i \in \mathbb{Z}_N$. Next, the challenger outputs 0 if $\overline{u}_i \notin \mathbb{NR}_{2N} \times \mathbb{H}$. Otherwise, it writes $\overline{u}_i$ as $g^{u_i} h^{\hat{u}_i}$ for $u_i \in \mathbb{Z}_{p'q'}$ and $\hat{u}_i \in \mathbb{Z}_N$. Then the challenger does the following:
  - If $r_i \neq H(\overline{t}_i^2)$, output 0.
  - If $\hat{c} \neq 0$ or $\hat{t}_i \neq 0$, then output 0.
  - If $t_i \neq s_i c$, then output 0.
  - Otherwise, take any $\mathbf{y} \in \mathbb{Z}^{\rho}_{p'q'}$ such that $\mathbf{y}^\mathsf{T}\mathbf{v} = c$. Output 1 if $u_i = (as_i + b_i)\mathbf{y}^\mathsf{T}\mathbf{v}$ and $\hat{u}_i = 0$. Otherwise (or if no such $\mathbf{y}$ exists), output 0.

  Importantly, the challenger's responses to the verification queries in Hyb_2 depend *only* on the public components (i.e., crs and pk).

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of an execution of experiment Hyb_i with adversary $\mathcal{A}$. We now show that the output distribution of each adjacent pair of hybrid experiments is statistically indistinguishable.

Lemma E.9. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. The statistical distance between $\mathsf{Uniform}(\mathbb{Z}_{\lfloor N^2/4 \rfloor})$ and $\mathsf{Uniform}(\mathbb{Z}_{Np'q'})$ satisfies

$$\Delta(\mathsf{Uniform}(\mathbb{Z}_{\lfloor N^2/4 \rfloor}), \mathsf{Uniform}(\mathbb{Z}_{Np'q'})) = 1 - \frac{Np'q'}{\lfloor N^2/4 \rfloor} = \mathsf{negl}(\lambda), \tag{E.1}$$

since $1/p', 1/q' = \mathsf{negl}(\lambda)$. Since $\rho = \mathsf{poly}(\lambda)$, the claim follows by a union bound. □

Lemma E.10. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. In Hyb_1 and Hyb_2, the challenger evaluates Verify at most $Q + 1$ times where $Q = \mathsf{poly}(\lambda)$ is the bound on the number of queries the adversary makes. For $j \in \{0, \ldots, Q+1\}$, let $\mathsf{Hyb}_{1,j}$ denote the experiment where the first j queries are handling according to the specification in Hyb_2 while the remaining queries are handling according to the specification in Hyb_1. By construction, $\mathsf{Hyb}_1 \equiv \mathsf{Hyb}_{1,0}$ and $\mathsf{Hyb}_2 \equiv \mathsf{Hyb}_{1,Q+1}$. Consider $\mathsf{Hyb}_{1,j-1}$ and $\mathsf{Hyb}_{1,j}$ for $j \in [Q+1]$. These two experiments only differ in how the challenger computes the output for the j^{th} Verify call. Moreover, by construction of $\mathsf{Hyb}_{1,j-1}$ and $\mathsf{Hyb}_{1,j}$, all of the adversary's queries prior to the j^{th} query are handled according to the specification in Hyb_2, which depend only on the public components crs and pk. Let (σ, i, r_i, π_i) be the arguments to the j^{th} Verify call. As in the specification of Hyb_2, parse $\sigma = \overline{c} \in \mathbb{Z}^*_{N^2}$ and $\pi_i = (\overline{t}_i, \overline{u}_i) \in \mathbb{Z}^*_{N^2}$. First, if $\overline{u}_i$ is not a quadratic residue (i.e., $\overline{u}_i \notin \mathbb{NR}_{2N} \times \mathbb{H}$), then the challenger in $\mathsf{Hyb}_{1,j}$ always outputs 0. We argue that this is the case in $\mathsf{Hyb}_{1,j-1}$ as well. By construction of Verify, the output is 1 only if $\overline{u}_i = (\overline{t}_i^a \overline{c}^{b_i})^2$. But if $\overline{u}_i$ is not a quadratic residue, this relation is never satisfied, and so the output in $\mathsf{Hyb}_{1,j}$ is also 0. Thus, it suffices to only consider the case where $\overline{u}_i$ is a quadratic residue. Then, define $c, t_i, u_i \in \mathbb{Z}_{p'q'}$ and $\hat{c}, \hat{t}_i, \hat{u}_i \in \mathbb{Z}_N$ such that $\overline{c}^2 = g^c h^{\hat{c}}$, $\overline{t}_i^2 = g^{t_i} h^{\hat{t}_i}$, and $\overline{u}_i = g^{u_i} h^{\hat{u}_i}$. We consider the different cases:

- Suppose $r_i \neq H(\overline{t}_i^2)$. Then, the output in both experiments is 0.

- Suppose $\hat{c} \neq 0$ or $\hat{t}_i \neq 0$. We argue that with overwhelming probability, the output in $\mathsf{Hyb}_{1,j-1}$ is 0. For the output to be 1 in $\mathsf{Hyb}_{1,j-1}$, it must be the case that $\overline{u}_i = (\overline{t}_i^a \overline{c}^{b_i})^2$, or equivalently,

$$u_i = at_i + b_i c \in \mathbb{Z}_{p'q'} \quad \text{and} \quad \hat{u}_i = a\hat{t}_i + b_i\hat{c} \in \mathbb{Z}_N.$$

  In both Hyb_1 and Hyb_2, the public parameters are *independent* of the values of $a \bmod N$ and $b_i \bmod N$. This follows from the fact that $a, b_i \in \mathbb{Z}_{Np'q'}$, and the adversary only sees elements $g^{(as_i+b_i)\mathbf{v}}$, where g generates a group of order $p'q'$ (and $\gcd(p'q', N) = 1$). Next, since a and b_i are uniform over $\mathbb{Z}_{Np'q'}$, the values $a \bmod N$ and $b_i \bmod N$ are distributed uniformly and independently of the rest of the public parameters. Since the responses to all of the adversary's queries prior to its j^{th} query only depend on the public parameters, the conditional distribution of $a \bmod N$ and $b_i \bmod N$ given the adversary's view up to the time of its j^{th} query is uniform and independently random. Since at least one of $\hat{c}$ and $\hat{t}_i$ is non-zero, this means that the value of $a\hat{t}_i + b_i\hat{c} \bmod N$ is independently and uniformly random over $\mathbb{Z}_N$. Therefore, $\hat{u}_i = a\hat{t}_i + b_i\hat{c}$ with probability at most $1/N = \mathsf{negl}(\lambda)$. This means that the output in $\mathsf{Hyb}_{1,j-1}$ is 0 with overwhelming probability.

- Suppose that $t_i \neq s_i c \in \mathbb{Z}_{p'q'}$. We only need to consider the case where $\hat{c} = 0$ and $\hat{t}_i = 0$. We show that in this case, the output in $\mathsf{Hyb}_{1,j-1}$ is 0 with overwhelming probability. For the output to be 1 in $\mathsf{Hyb}_{1,j-1}$, it must be the case that $\overline{u}_i = (\overline{t}_i^a \overline{c}^{b_i})^2$, or equivalently,

$$u_i = at_i + b_i c \in \mathbb{Z}_{p'q'} \quad \text{and} \quad \hat{u}_i = a\hat{t}_i + b_i\hat{c} = 0 \in \mathbb{Z}_N.$$

  The only components in crs and pk that depend on a and b_i are the public-key components $g^{(as_i+b_i)\mathbf{v}}$. Let $\mathbf{b} \in \mathbb{Z}^{\rho}_{Np'q'}$ be the vector whose components are $b_1, \ldots, b_\rho$, and let $\mathbf{s} \in \mathbb{Z}^{\rho}_{Np'q'}$ be the vector whose components are $s_1, \ldots, s_\rho$. The public parameters pk can then be expressed as a function of

$$\mathbf{Z} = [\ \mathbf{s} \mid \mathbf{I}_\rho\ ] \cdot \begin{bmatrix} a \\ \mathbf{b} \end{bmatrix} \cdot \mathbf{v}^{\mathsf{T}} \in \mathbb{Z}^{\rho\times\rho}_{Np'q'},$$

  where $\mathbf{I}_\rho \in \mathbb{Z}^{\rho\times\rho}_{Np'q'}$ is the identity matrix. Namely, the components of pk consist of $g^{\mathbf{Z}}$. Since $t_i \neq s_i c \in \mathbb{Z}_{p'q'}$, by the Chinese remainder theorem, it must be the case that $t_i \neq s_i c \pmod{p'}$ or $t_i \neq s_i c \pmod{q'}$. Without loss of generality, suppose that $t_i \neq s_i c \pmod{p'}$. In this case the vector $[\ t_i \mid c \cdot \mathbf{e}_i\ ]$ is linearly independent of the rows of the matrix $[\ \mathbf{s} \pmod{p'} \mid \mathbf{I}_\rho\ ]$. Since $a, \mathbf{b}$ are uniform over $\mathbb{Z}_{Np'q'}$, the components $a \bmod p'$ and $\mathbf{b} \bmod p'$ are uniform over $\mathbb{Z}_{p'}$. By linear independence over $\mathbb{Z}_{p'}$, the value of $at_i + b_i c \pmod{p'}$ is uniformly random over $\mathbb{Z}_{p'}$ even given $\mathbf{Z}$. This means that $at_i + b_i c = u_i \pmod{p)'}$ with probability at most $1/p' = \mathsf{negl}(\lambda)$. Thus, the challenger in $\mathsf{Hyb}_{1,j-1}$ outputs 0 with overwhelming probability.

- The only remaining case is when $\hat{c} = 0 = \hat{t}_i$, $t_i = s_i c$, and $r_i = H(\overline{t}_i^2)$. In this case, $(g^{t_i})^a (g^c)^{b_i} = g^{(as_i+b_i)c}$. Since $\mathbf{v} \xleftarrow{\text{R}} \mathbb{Z}_{\lfloor N^2/4 \rfloor}$, with overwhelming probability, there will be some component that is invertible modulo $p'q'$. If so, there always exists $\mathbf{y} \in \mathbb{Z}_{p'q'}$ such that $\mathbf{y}^{\mathsf{T}}\mathbf{v} = c$. Then, in $\mathsf{Hyb}_{1,j-1}$, the challenger outputs 1 if and only if

$$\overline{u}_i = (\overline{t}_i^a \overline{c}^{b_i})^2 = g^{at_i+b_ic} = g^{(as_i+b_i)c} = g^{(as_i+b_i)\mathbf{y}^{\mathsf{T}}\mathbf{v}}.$$

  Since $\overline{u}_i = g^{u_i} h^{\hat{u}_i}$, this is equivalent to checking that $u_i = (as_i + b_i)\mathbf{y}^{\mathsf{T}}\mathbf{v}$ and $\hat{u}_i = 0$. This is precisely the check in $\mathsf{Hyb}_{1,j}$.

In each case, we see that the j^{th} call to Verify is implemented correctly with overwhelming probability in $\mathsf{Hyb}_{1,j-1}$ and $\mathsf{Hyb}_{1,j}$. □

To complete the proof, it suffices to show that for all adversaries $\mathcal{A}$, the output of $\mathsf{Hyb}_2(\mathcal{A})$ is 0 with overwhelming probability. Let $(\sigma^*, i^*, r^*, \pi^*)$ be the adversary's output in Hyb_2. The output of Hyb_2 is 1 only if $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma^*, i^*, r^*, \pi^*) = 1$ and $r_{i^*} \neq r^*$ where $r \leftarrow \mathsf{Open}(\mathsf{crs}, \sigma^*)$. Write $\sigma^* = \bar{c}$, $\bar{c}^2 = g^c h^{\hat{c}}$, $\pi^* = (\bar{t}, \bar{u})$, $\bar{t}^2 = g^t h^{\hat{t}}$, and $\bar{u} = g^u h^{\hat{u}}$ (if $\bar{u}$ is not a quadratic residue in $\mathbb{Z}_{N^2}$, then the output in Hyb_2 is 0). In Hyb_2, if $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma^*, i^*, r^*, \pi^*) = 1$, it must be the case that

$$r^* = H(\bar{t}^2) \quad \text{and} \quad \hat{c} = 0 \quad \text{and} \quad \hat{t} = 0 \quad \text{and} \quad t = s_{i^*} c.$$

Consider now the value of r_{i^*} and r^*:

- From the above, $\bar{t}^2 = g^t h^{\hat{t}} = g^t = g^{c s_{i^*}}$. Thus $r^* = H(\bar{t}^2) = H(g^{c s_{i^*}})$.
- By definition of Open, $r_{i^*} = H((\bar{c}^{s_{i^*}})^2) = H((\bar{c}^2)^{s_{i^*}}) = H((g^c)^{s_{i^*}}) = r^*$.

Since $r^* = r_{i^*}$, the output in $\mathsf{Hyb}_2(\mathcal{A})$ is 0. □

Proof of Theorem E.7 (Statistical Simulation). We construct a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ as follows:

- $\mathcal{S}_1(1^\lambda, 1^\rho) \to (\mathsf{st}_{\mathcal{S}}, \widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}})$: Sample $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$, where $p = 2p' + 1$ and $q = 2q' + 1$. Let g be a generator of $\mathbb{NR}_{2N}$ and $h = 1 + N$ be the generator of $\mathbb{H}$. Sample a vector $\mathbf{v} \xleftarrow{\text{R}} \mathbb{Z}^\rho_{\lfloor N^2/4 \rfloor}$, scalars $s_1, \ldots, s_\rho \xleftarrow{\text{R}} \mathbb{Z}_{\lfloor N^2/4 \rfloor}$, and a hash function $H \xleftarrow{\text{R}} \mathcal{H}$. Sample $a, b_1, \ldots, b_\rho \xleftarrow{\text{R}} \mathbb{Z}_{\lfloor N^2/4 \rfloor}$, and set

$$\begin{aligned}
\widetilde{\mathsf{crs}} &= \left(N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\mathbf{e}_1}, \ldots, g^{s_\rho \mathbf{v}} h^{\mathbf{e}_\rho}\right) \\
\widetilde{\mathsf{pk}} &= \left(g^{(a s_1 + b_1)\mathbf{v}} h^{a \mathbf{e}_1}, \ldots, g^{(a s_\rho + b_\rho)\mathbf{v}} h^{a \mathbf{e}_\rho}\right) \\
\widetilde{\mathsf{sk}} &= \left(a, b_1, \ldots, b_\rho\right).
\end{aligned}$$

  Output $\widetilde{\mathsf{crs}}$, $\widetilde{\mathsf{pk}}$, $\widetilde{\mathsf{sk}}$ and $\mathsf{st}_{\mathcal{S}} = (\widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, p', q', s_1, \ldots, s_\rho)$.

- $\mathcal{S}_2(\mathsf{st}_{\mathcal{S}}, I, r_I) \to (\widetilde{\sigma}, \{\widetilde{\pi}_i\}_{i \in I})$: On input $\mathsf{st}_{\mathcal{S}} = (\widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, p', q', s_1, \ldots, s_\rho)$ where

$$\widetilde{\mathsf{crs}} = \left(N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\mathbf{e}_1}, \ldots, g^{s_\rho \mathbf{v}} h^{\mathbf{e}_\rho}\right) \quad \text{and} \quad \widetilde{\mathsf{pk}} = \left(g^{\mathbf{z}_1} h^{\hat{\mathbf{z}}_1}, \ldots, g^{\mathbf{z}_\rho} h^{\hat{\mathbf{z}}_\rho}\right),$$

  where $\mathbf{z}_i = (a s_i + b_i)\mathbf{v}$ and $\hat{\mathbf{z}}_i = a \mathbf{e}_i$ for all $i \in [\rho]$, a set of indices $I \subseteq [\rho]$, and a bitstring $r_I \in \{0,1\}^{|I|}$, the simulator samples $\mathbf{y}' \xleftarrow{\text{R}} \mathbb{Z}^\rho_{p'q'}$. Then, it samples a vector $\hat{\mathbf{y}}' \in \mathbb{Z}^\rho_N$ component-by-component: specifically, for each $i \in [\rho]$, it does the following:

  - If $i \notin I$, sample $\hat{y}'_i \xleftarrow{\text{R}} \mathbb{Z}_N$.
  - If $i \in I$, sample $\hat{y}'_i \xleftarrow{\text{R}} \mathbb{Z}_N$ conditioned on $r_i = H\left((g^{s_i (\mathbf{y}')^\intercal \mathbf{v}} h^{\hat{y}'_i})^2\right)$. Specifically, repeatedly sample $\hat{y}'_i \xleftarrow{\text{R}} \mathbb{Z}_N$ until finding one that satisfies the relation. If no candidate is found after λ attempts, then abort and output $\perp$.

Define $\mathbf{y} \in \mathbb{Z}^{\rho}_{Np'q'}$ to be the vector where $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod{N}$. The simulator then sets $\overline{t}_i = g^{s_i \mathbf{y}^\top \mathbf{v}} h^{\mathbf{y}^\top \mathbf{e}_i}$ and $\overline{u}_i = (g^{\mathbf{y}^\top \mathbf{z}_i} h^{\mathbf{y}^\top \hat{\mathbf{z}}_i})^2$. Output $\widetilde{\sigma} = g^{\mathbf{y}^\top \mathbf{v}}$ and $\{\widetilde{\pi}_i\}_{i \in I}$ where $\widetilde{\pi}_i = (\overline{t}_i, \overline{u}_i)$.

To show that $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 0]$ and $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 1]$ are statistically indistinguishable, we use a hybrid argument:

- Hyb_0: This is the experiment $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 0]$. Namely, the challenger begins by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs})$. For each challenge query, the challenger first samples $(\sigma, r, \{\pi_i\}_{i \in [\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs})$ and gives r to $\mathcal{A}$ before receiving a set $I \subseteq [\rho]$ chosen by $\mathcal{A}$. It then replies with σ and $\{\pi_i\}_{i \in I}$.

  More precisely, in this experiment,

  $$\mathsf{crs} = \big(N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\mathbf{e}_1}, \ldots, g^{s_\rho \mathbf{v}} h^{\mathbf{e}_\rho}\big) \quad \text{and} \quad \mathsf{pk} = \big(g^{\mathbf{z}_1} h^{\hat{\mathbf{z}}_1}, \ldots, g^{\mathbf{z}_\rho} h^{\hat{\mathbf{z}}_\rho}\big),$$

  where $\mathbf{z}_i = (as_i + b_i)\mathbf{v}$ and $\hat{\mathbf{z}}_i = a\mathbf{e}_i$ for all $i \in [\rho]$. On each challenge query, the challenger samples $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{\lfloor N^2/4 \rfloor}$ and computes

  $$\sigma = g^{\mathbf{y}^\top \mathbf{v}} \quad \text{and} \quad \overline{t}_i = g^{s_i \mathbf{y}^\top \mathbf{v}} h^{\mathbf{y}^\top \mathbf{e}_i} \quad \text{and} \quad \overline{u}_i = (g^{\mathbf{y}^\top \mathbf{z}_i} h^{\mathbf{y}^\top \hat{\mathbf{z}}_i})^2,$$

  for all $i \in [\rho]$. The random bits r_i satisfy $r_i = H(\overline{t}_i^2)$, and the proofs π_i satisfy $\pi_i = (\overline{t}_i, \overline{u}_i)$.

- Hyb_1: Same as Hyb_0, except that the challenger computes $(\mathsf{st}_{\mathcal{S}}, \widetilde{\mathsf{crs}}, \widetilde{\mathsf{pk}}, \widetilde{\mathsf{sk}}) \leftarrow \mathcal{S}_1(1^\lambda, 1^\rho)$ and uses $\widetilde{\mathsf{crs}}$, $\widetilde{\mathsf{pk}}$, and $\widetilde{\mathsf{sk}}$ instead of crs, pk, and sk, respectively.

- Hyb_2: Same as Hyb_1, except when responding to the challenge queries, the challenger samples $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{Np'q'}$ instead of $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{\lfloor N^2/4 \rfloor}$.

- Hyb_3: Same as Hyb_2, except when responding to the challenge queries, the challenger first samples $\mathbf{y}' \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{p'q'}$ and $\hat{\mathbf{y}}' \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{N}$. It defines $\mathbf{y} \in \mathbb{Z}^{\rho}_{Np'q'}$ to be the vector where $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod{N}$.

- Hyb_4: Same as Hyb_3, except when responding to the challenge queries, the challenger first samples $r \xleftarrow{\mathrm{R}} \{0,1\}^\rho$. Then it samples $\mathbf{y}' \xleftarrow{\mathrm{R}} \mathbb{Z}^{\rho}_{p'q'}$. Next, for each $i \in [\rho]$, it samples $\hat{y}'_i \xleftarrow{\mathrm{R}} \mathbb{Z}_N$ such that $r_i = H\big((g^{s_i (\mathbf{y}')^\top \mathbf{v}} h^{\hat{y}'_i})^2\big)$. It uses the same rejection sampling procedure as in $\mathcal{S}_2$: namely, repeated sample $\hat{y}'_i \xleftarrow{\mathrm{R}} \mathbb{Z}_N$ until finding one that satisfies the relation, and abort with output $\perp$ if no such $\hat{y}'_i$ is found after λ attempts. Finally it sets $\mathbf{y} \in \mathbb{Z}^{\rho}_{Np'q'}$ to be the vector where $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod{N}$. The remaining components are constructed as before.

- Hyb_5: Same as Hyb_4, except when responding to the challenge queries, the challenger samples $\hat{\mathbf{y}}' \in \mathbb{Z}^{\rho}_{N}$ *after* it receives the challenge set. In particular, on each query, after the challenger receives the set $I \subseteq [\rho]$, for each $i \in I$, it applies the same rejection sampling procedure as $\mathcal{S}_2$ to sample $\hat{y}'_i \in \mathbb{Z}_N$ such that $r_i = H\big((g^{s_i (\mathbf{y}')^\top \mathbf{v}} h^{\hat{y}'_i})^2\big)$. If $i \notin I$, then it samples $r_i \xleftarrow{\mathrm{R}} \mathbb{Z}_N$. All remaining components are constructed as in Hyb_4. This is the distribution in $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 1]$.

Lemma E.11. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \equiv \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. Since $\mathcal{S}_1(1^\lambda, 1^\rho)$ samples $\widetilde{\mathsf{crs}}$, $\widetilde{\mathsf{pk}}$, and $\widetilde{\mathsf{sk}}$ using the same procedure as Setup and KeyGen, the output distributions of hybrids Hyb_0 and Hyb_1 are identically distributed. □

Lemma E.12. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. The only difference between Hyb_1 and Hyb_2 is that the challenger samples $\mathbf{y}$ uniformly at random from $\mathbb{Z}^\rho_{Np'q'}$ instead of $\mathbb{Z}^\rho_{\lfloor N^2/4 \rfloor}$ when answering challenge queries. Since the distributions $\mathsf{Uniform}(\mathbb{Z}_{\lfloor N^2/4 \rfloor})$ and $\mathsf{Uniform}(\mathbb{Z}_{Np'q'})$ are statistically indistinguishable (see Eq. (E.1)), the claim follows by a union bound (since $\rho = \mathsf{poly}(\lambda)$ and $\mathcal{A}$ makes a polynomial number of queries). □

Lemma E.13. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_2(\mathcal{A}) \equiv \mathsf{Hyb}_3(\mathcal{A})$.*

Proof. The two distributions only differ in how $\mathbf{y}$ is sampled. In Hyb_2, $\mathbf{y}$ is uniform over $\mathbb{Z}^\rho_{Np'q'}$ while in Hyb_3, we sample $\mathbf{y}' \xleftarrow{\mathrm{R}} \mathbb{Z}^\rho_{p'q'}$ and $\hat{\mathbf{y}}' \xleftarrow{\mathrm{R}} \mathbb{Z}^\rho_N$ and define $\mathbf{y}$ so that $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod{N}$. These distributions are identical by the Chinese remainder theorem (since $\gcd(p'q', N) = 1$). □

Lemma E.14. *If $\mathcal{H}$ is a universal hash, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_3(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_4(\mathcal{A})$.*

Proof. In Hyb_3 and Hyb_4, the challenger first samples $\mathbf{y}' \xleftarrow{\mathrm{R}} \mathbb{Z}^\rho_{p'q'}$. In Hyb_3, the challenger then samples $\hat{y}'_i \xleftarrow{\mathrm{R}} \mathbb{Z}_N$, and sets

$$r_i = H((g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i})^2) \tag{E.2}$$

for each $i \in [\rho]$. In Hyb_4, the challenger first samples $r_i \xleftarrow{\mathrm{R}} \{0,1\}$ and then samples $\hat{y}'_i \xleftarrow{\mathrm{R}} \mathbb{Z}^\rho_N$ such that Eq. (E.2) holds for each $i \in [\rho]$. First, all of the other components in Eq. (E.2) (i.e., s_i, $\mathbf{v}$, and $\mathbf{y}'$) are *identically* distributed in Hyb_3 and Hyb_4). Thus, it suffices to show the following:

- The distribution of r_i in Hyb_3 is statistically close to uniform over $\{0,1\}$. This follows from the fact that $\hat{y}'_i$ is uniform over $\mathbb{Z}_N$. This means that $\hat{y}'_i$ is sampled from a distribution with at least $\log N$ bits of min-entropy. Since $\gcd(N, 2) = 1$, the distribution of $(g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i})^2$ also has at least $\log N$ bits of min-entropy over $\mathbb{Z}_{N^2}$ (even after fixing $s_i, \mathbf{v}, \mathbf{y}'$). Since $\mathcal{H}$ is universal, $1/N = \mathsf{negl}(\lambda)$, we appeal to the the leftover hash lemma to conclude that r_i is statistically close to uniform over $\{0,1\}$. Since each $\hat{y}'_i$ is sampled independently, each r_i is correspondingly independent and statistically close to uniform. By a union bound, we conclude that $r \in \{0,1\}^\lambda$ is statistically close to uniform in Hyb_3. Thus, the distribution of r in the two experiments are statistically indistinguishable.

- With overwhelming probability, the sampling algorithm in Hyb_4 does not abort. If this is the case, then in Hyb_4, the distribution of $\hat{y}'_i$ is uniform over $\mathbb{Z}_N$ subject to Eq. (E.2) holding, which coincides with the distribution in Hyb_3. From the above analysis, we have that for $\hat{y}'_i \xleftarrow{\mathrm{R}} \mathbb{Z}_N$, the distribution of $H((g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_i})^2)$ is statistically close to uniform over $\{0,1\}$, and thus, will equal r_i with probability at least $1/2 - \mathsf{negl}(\lambda)$. The probability that the challenger fails to sample a $\hat{y}'_i$ such that Eq. (E.2) holds is then $1/2^\lambda - \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda)$. Thus, the challenger in Hyb_4 succeeds with overwhelming probability, and the claim follows. □

Lemma E.15. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_4(\mathcal{A}) \equiv \mathsf{Hyb}_5(\mathcal{A})$.*

Proof. First, the challenger generates the common reference string crs and the public key pk identically in the two experiments. It suffices to argue that the challenger's response to the adversary's challenge query is identically distributed in Hyb_4 and Hyb_5. In particular, on each challenge query, after the adversary outputs a set $I \subseteq [\rho]$, the challenger replies with a commitment $\sigma = g^{\mathbf{y}^\mathsf{T}\mathbf{v}}$ and a collection of proofs $\{\pi_i\}_{i \in I}$ where $\pi_i = (\bar{t}_i, \bar{u}_i)$. We show that the commitment σ and the group elements $\bar{t}_i, \bar{u}_i \in \mathbb{Z}_N^*$ are identically distributed in the two experiments:

- In Hyb_4 and Hyb_5, the challenger samples $\mathbf{y}' \xleftarrow{\mathrm{R}} \mathbb{Z}_{p'q'}^{\rho}$. Since g generates a subgroup of order $p'q'$ and $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$, it follows that $\sigma = g^{\mathbf{y}^\mathsf{T}\mathbf{v}} = g^{(\mathbf{y}')^\mathsf{T}\mathbf{v}}$, and so σ is identically distributed in the two experiments.
- In Hyb_4 and Hyb_5, for all $i \in I$, the challenger samples $\hat{y}_i' \in \mathbb{Z}_2$ conditioned on $r_i = H\big((g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}_i'})^2\big)$. Thus, the variables $\hat{y}_i'$ for $i \in I$ in the two experiments are identically distributed. In both experiments, the value $\bar{t}_i$ satisfies $\bar{t}_i = g^{s_i(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}_i'}$, and since $\mathbf{y}'$ and $\hat{y}_i'$ are identically distributed in the two experiments (along with the remaining components s_i, and $\mathbf{v}$), the value $\bar{t}_i$ is also identically distributed in the two experiments.
- In Hyb_4 and Hyb_5, $\bar{u}_i = (g^{\mathbf{y}^\mathsf{T}\mathbf{z}_i} h^{\mathbf{y}^\mathsf{T}\hat{\mathbf{z}}_i})^2$, where $\mathbf{z}_i = (as_i + b_i)\mathbf{v}$ and $\hat{\mathbf{z}}_i = a\mathbf{e}_i$. In particular, this means that $\bar{u}_i = (g^{(as_i+b_i)(\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{a\hat{y}_i'})^2$. By construction, for all $i \in I$, all of the terms in the exponents are identically distributed in the two experiments. □

The theorem now follows by a hybrid argument. □

E.2 Dual-Mode Hidden-Bits Generator with Malicious Security from DCR

In this section, we describe our construction of a dual-mode hidden-bits generator with malicious security from the decisional composite residuosity (DCR) assumption from [Pai99]. This is an analog of Construction 5.9 from the QR assumption.

Construction E.16 (Dual-Mode HBG with Malicious Security from DCR)**.** Let ρ be the output length of the hidden-bits generator. Our construction relies on a similar set of building blocks as Construction 4.18:

- Let $\mathsf{SampleModulus}$ be a safe prime modulus sampler.
- Let $\ell = 3\rho\lambda$ and define $\mathcal{T}_{\lambda,\ell} := \{S \subseteq [\ell] : |S| = \lambda\}$ to be the set of all subsets of $[\ell]$ that contains exactly λ elements. Let $G\colon \{0,1\}^\kappa \to \mathcal{T}_{\lambda,\ell}^{\rho} \times \mathbb{Z}_N^{\rho\ell}$ be a PRG with seed length $\kappa = \kappa(\lambda)$. Here, N is the modulus output by $\mathsf{SampleModulus}$. We refer to Construction 4.18 for a description of how to construct such a PRG.

We construct the dual-mode designated-verifier HBG with malicious security as follows:

- $\mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{mode}) \to \mathsf{crs}$: Let $\ell' = \rho\ell$. Sample $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$. Let g be a generator of $\mathbb{NR}_{2N}$ and $h = 1 + N$ be the generator of $\mathbb{H}$. The setup algorithm samples a vector $\mathbf{v} \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N^2/4 \rfloor}^{\ell'}$, scalars $s_1, \ldots, s_{\ell'} \xleftarrow{\mathrm{R}} \mathbb{Z}_{\lfloor N^2/4 \rfloor}$ and sets $\hat{\mathbf{w}}_i \in \mathbb{Z}_N^{\ell'}$ for $i \in [\ell']$ as follows:
  - if $\mathsf{mode} = \mathsf{hiding}$, set $\hat{\mathbf{w}}_i \leftarrow \mathbf{e}_i$, where $\mathbf{e}_i \in \mathbb{Z}_N^{\ell'}$ is the i^{th} basis vector.
  - If $\mathsf{mode} = \mathsf{binding}$, set $\hat{\mathbf{w}}_i \leftarrow \mathbf{0}$.

Finally, it samples a hash function $H \xleftarrow{R} \mathcal{H}$ where $\mathcal{H}$ is a family of hash functions with domain $\mathbb{Z}_{N^2}$ and range $\{0,1\}$. Output $\mathsf{crs} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\hat{\mathbf{w}}_1}, \ldots, g^{s_{\ell'} \mathbf{v}} h^{\hat{\mathbf{w}}_{\ell'}})$.

- $\mathsf{KeyGen}(\mathsf{crs}) \to (\mathsf{pk}, \mathsf{sk})$: On input $\mathsf{crs} = \left(N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_{\ell'}\right)$, sample $a, b_1, \ldots, b_{\ell'} \xleftarrow{R} \mathbb{Z}_{\lfloor N^2/4 \rfloor}$ for each $i \in [\ell']$. Output $\mathsf{pk} = (\overline{\mathbf{v}}^{b_1} \overline{\mathbf{w}}_1^a, \ldots, \overline{\mathbf{v}}^{b_{\ell'}} \overline{\mathbf{w}}_{\ell'}^a)$ and $\mathsf{sk} = (a, b_1, \ldots, b_{\ell'})$.

- $\mathsf{GenBits}(\mathsf{crs}, \mathsf{pk}) \to (\sigma, r, \{\pi_i\}_{i \in [\rho]})$: On input $\mathsf{crs} = \left(N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_{\ell'}\right)$ and $\mathsf{pk} = (\overline{\mathbf{z}}_1, \ldots, \overline{\mathbf{z}}_{\ell'})$, first check that $\overline{\mathbf{z}}_i \in (\mathbb{Z}_{N^2}^*)^{\ell'}$ for all $i \in [\ell']$ (and output $\perp$ otherwise). Then, sample $\mathbf{y} \xleftarrow{R} \mathbb{Z}_{\lfloor N^2/4 \rfloor}^{\ell'}$ and compute for each $i \in [\ell']$:

$$\overline{c} \leftarrow \prod_{j \in [\ell']} \overline{v}_j^{y_j} \quad \text{and} \quad \overline{t}_i \leftarrow \prod_{j \in [\ell']} \overline{w}_{i,j}^{y_j} \quad \text{and} \quad \overline{u}_i \leftarrow \prod_{j \in [\ell']} (\overline{z}_{i,j}^2)^{y_j}.$$

  Next, sample a PRG seed $\mathsf{s} \xleftarrow{R} \{0,1\}^\kappa$ and compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$ where $\hat{S}_i \in \mathcal{T}_{\lambda,\ell}$ for all $i \in [\rho]$ and $\boldsymbol{\alpha} \in \mathbb{Z}_N^{\rho\ell}$. Compute the shifted sets $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$ for each $i \in [\rho]$. Finally, compute

$$r_i \leftarrow H\left(\prod_{j \in S_i} \overline{t}_j^{2\alpha_j}\right) \quad \text{and} \quad \pi_i \leftarrow \{(j, \overline{t}_j, \overline{u}_j)\}_{j \in S_i}.$$

  Output $\sigma = (\mathsf{s}, \overline{c})$, r, and $\{\pi_i\}_{i \in [\rho]}$.

- $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \sigma, i, r_i, \pi_i)$: On input $\mathsf{crs} = \left(N, g, h, H, \overline{\mathbf{v}}, \overline{\mathbf{w}}_1, \ldots, \overline{\mathbf{w}}_{\ell'}\right)$, $\mathsf{sk} = (a, b_1, \ldots, b_{\ell'})$, $\sigma = (\mathsf{s}, \overline{c})$, $i \in [\rho]$, $r_i \in \{0,1\}$, and $\pi_i = \{(j, \overline{t}_j, \overline{u}_j)\}_{j \in S}$ for an implicitly-defined set $S \subseteq [\ell']$, the verification algorithm performs the following checks:
  - Compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$ and the shifted set $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$. It checks that $S = S_i$ and outputs 0 if not.
  - It checks that $\overline{u}_j = (\overline{t}_j^a \overline{c}^{b_j})^2$ for all $j \in S$, and outputs 0 if not.
  - It checks that $r_i = H\big(\prod_{j \in S} \overline{t}_j^{2\alpha_j}\big)$ and outputs 0 if not.

  If all checks pass, the verification algorithm outputs 1.

Correctness and security analysis. We state the correctness and security theorems for Construction 5.3 here, but defer the proofs to Appendix E.2.1.

Theorem E.17 (Correctness)**.** *Construction E.16 is correct.*

Theorem E.18 (Succinctness)**.** *Construction E.16 is succinct.*

Theorem E.19 (CRS Indistinguishability)**.** *Suppose the* DCR *assumption holds with respect to* SampleModulus*. Then, Construction E.16 satisfies CRS indistinguishability.*

Theorem E.20 (Statistical Binding in Binding Mode)**.** *Construction E.16 satisfies statistical binding in binding mode.*

Theorem E.21 (Statistical Simulation in Hiding Mode)**.** *Construction E.16 satisfies statistical simulation in hiding mode.*

E.2.1 Analysis of Construction E.16 (Dual-Mode (Malicious) HBG from DCR)

In this section, we give the proofs for the correctness and security theorems (Theorems E.17 to E.21) for the dual-mode hidden-bits generator with security against malicious verifiers from the DCR assumption (Construction E.16). The analysis is very similar to that of the basic scheme from DCR (Construction E.2).

Proof of Theorem E.17 (Correctness). Follows by an analogous argument as the proof of Theorem E.3. □

Proof of Theorem E.18 (Succinctness). The commitment σ in Construction E.16 consists of a PRG seed $\mathsf{s} \in \{0,1\}^\kappa$ where $\kappa = \mathsf{poly}(\lambda)$ and an element of $\mathbb{Z}_{N^2}$, which has size $2\lceil \log N \rceil = \mathsf{poly}(\lambda)$. □

Proof of Theorem E.19 (CRS Indistinguishability). Same as the proof of Theorem E.5. □

Proof of Theorem E.20 (Statistical Binding). Follows by a similar argument as the proof of Theorem E.6. □

Proof of Theorem E.21 (Statistical Hiding). We construct a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ as follows:

- $\mathcal{S}_1(1^\lambda, 1^\rho) \to (\mathsf{st}_{\mathcal{S}}, \widetilde{\mathsf{crs}})$: Sample $(N, p, q) \leftarrow \mathsf{SampleModulus}(1^\lambda)$, where $p = 2p' + 1$ and $q = 2q' + 1$. Let g be a generator of $\mathbb{NR}_{2N}$ and $h = 1 + N$ be the generator of $\mathbb{H}$. Sample a vector $\mathbf{v} \xleftarrow{R} \mathbb{Z}^{\ell'}_{\lfloor N^2/4 \rfloor}$, scalars $s_1, \ldots, s_{\ell'} \xleftarrow{R} \mathbb{Z}_{\lfloor N^2/4 \rfloor}$, and a hash function $H \xleftarrow{R} \mathcal{H}$. Output

$$\widetilde{\mathsf{crs}} = \big(N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\mathbf{e}_1}, \ldots, g^{s_{\ell'} \mathbf{v}} h^{\mathbf{e}_{\ell'}}\big).$$

  and $\mathsf{st}_{\mathcal{S}} = (\widetilde{\mathsf{crs}}, p', q', s_1, \ldots, s_{\ell'})$.

- $\mathcal{S}_2(\mathsf{st}_{\mathcal{S}}, \mathsf{pk}, I, r_I) \to (\widetilde{\sigma}, \{\widetilde{\pi}_i\}_{i \in I})$: On input $\mathsf{st}_{\mathcal{S}} = (\widetilde{\mathsf{crs}}, p', q', s_1, \ldots, s_{\ell'})$ where

$$\widetilde{\mathsf{crs}} = \big(N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\mathbf{e}_1}, \ldots, g^{s_{\ell'} \mathbf{v}} h^{\mathbf{e}_{\ell'}}\big),$$

  a public key $\mathsf{pk} = (\overline{\mathbf{z}}_1, \ldots, \overline{\mathbf{z}}_{\ell'})$, a set of indices $I \subseteq [\rho]$, and a bitstring $r_I \in \{0,1\}^{|I|}$, the simulator does the following:

  1. Check that $\overline{\mathbf{z}}_i \in (\mathbb{Z}^*_{N^2})^{\ell'}$ for all $i \in [\ell']$. Output $\perp$ if this is not the case.
  2. Sample a seed $\mathsf{s} \xleftarrow{R} \{0,1\}^\kappa$ and compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$, where $\boldsymbol{\alpha} \in \mathbb{Z}^{\rho\ell}_N$. For each $i \in I$, it computes the shifted sets $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$.
  3. Sample $\mathbf{y}' \xleftarrow{R} \mathbb{Z}^{\ell'}_{p'q'}$. Then, it samples a vector $\hat{\mathbf{y}}' \in \mathbb{Z}^{\ell'}_N$ as follows:
     - For each $i \in I$, let $\omega_i \leftarrow \sum_{j \in S_i} \alpha_j s_j \pmod{p'q'}$. Then, sample $\hat{\omega}_i \xleftarrow{R} \mathbb{Z}_N$ conditioned on $r_i = H\big((g^{\omega_i (\mathbf{y}')^\intercal \mathbf{v}} h^{\hat{\omega}_i})^2\big)$. Specifically, repeatedly sample $\hat{\omega}_i \xleftarrow{R} \mathbb{Z}_N$ until finding one that satisfies the relation. If no candidate is found after λ attempts, then abort and output $\perp$. After sampling $\hat{\omega}_i$, sample $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_N$ for each $j \in S_i$ subject to the constraint that $\sum_{j \in S_i} \alpha_j \hat{y}'_j = \hat{\omega}_i$.

– For all of the remaining indices $j \in [\ell'] \setminus \bigcup_{i \in I} S_i$, sample $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_N$.

Define $\mathbf{y} \in \mathbb{Z}^{\ell'}_{Np'q'}$ to be the vector where $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod{N}$.

4. Next, the simulator computes $\widetilde{\sigma} = (\mathsf{s}, g^{\mathbf{y}^\mathsf{T}\mathbf{v}})$, $\overline{t}_j = g^{s_j \mathbf{y}^\mathsf{T}\mathbf{v}} h^{\mathbf{y}^\mathsf{T}\mathbf{e}_j}$, and $\overline{u}_j \leftarrow \prod_{k \in [\ell']} (\overline{z}^2_{j,k})^{y_k}$ for all $j \in S_i$ and $i \in I$. It sets $\widetilde{\pi}_i = \{(j, \overline{t}_j, \overline{u}_j)\}_{j \in S_i}$.
5. Output $\widetilde{\sigma}$ and $\{\widetilde{\pi}_i\}_{i \in I}$.

We now use a hybrid argument to show that $\mathsf{ExptHide}[\mathcal{A}, 0]$ and $\mathsf{ExptHide}[\mathcal{A}, 1]$ are statistically indistinguishable:

- Hyb_0: This is the experiment $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 0]$. Namely, the challenger begins by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\rho, \mathsf{hiding})$, and gives crs to $\mathcal{A}$. The adversary $\mathcal{A}$ replies with a public key pk. For each challenge query, the challenger samples $(\sigma, r, \{\pi_i\}_{i \in [\rho]}) \leftarrow \mathsf{GenBits}(\mathsf{crs}, \mathsf{pk})$ and gives r to $\mathcal{A}$ before receiving a set $I \subseteq [\rho]$ chosen by $\mathcal{A}$. It then replies with σ and $\{\pi_i\}_{i \in I}$.

- Hyb_1: This experiment is identical to Hyb_0, except that the challenger computes $(\mathsf{st}_\mathcal{S}, \widetilde{\mathsf{crs}}) \leftarrow \mathcal{S}_1(1^\lambda, 1^\rho)$ and uses $\widetilde{\mathsf{crs}}$ in place of crs. Everything else proceeds identically to Hyb_0.

  Specifically, in this experiment, the challenger samples (N, p, q), H, $\mathbf{v}$, $s_1, \ldots, s_{\ell'}$ as specified by $\mathcal{S}_1$ and sets $\widetilde{\mathsf{crs}} = (N, g, h, H, g^{\mathbf{v}}, g^{s_1 \mathbf{v}} h^{\mathbf{e}_1}, \ldots, g^{s_{\ell'} \mathbf{v}} h^{\mathbf{e}_{\ell'}})$. It gives $\widetilde{\mathsf{crs}}$ to $\mathcal{A}$ to receive a public key $\mathsf{pk} = (\overline{\mathbf{z}}_1, \ldots, \overline{\mathbf{z}}_{\ell'})$. On a challenge query, the challenger proceeds as follows:

  1. Check that $\overline{\mathbf{z}}_i \in (\mathbb{Z}^*_{N^2})^{\ell'}$ for all $i \in [\ell']$, and output $\perp$ otherwise.
  2. Sample $\mathsf{s} \xleftarrow{R} \{0,1\}^\kappa$ and compute $(\hat{S}_1, \ldots, \hat{S}_\rho, \boldsymbol{\alpha}) \leftarrow G(\mathsf{s})$. For each $i \in [\rho]$, let $S_i \leftarrow \{j + \ell \cdot (i-1) \mid j \in \hat{S}_i\}$.
  3. Sample $\mathbf{y} \xleftarrow{R} \mathbb{Z}^{\ell'}_{\lfloor N^2/4 \rfloor}$ and compute for each $j \in [\ell']$,

  $$\overline{c} \leftarrow g^{\mathbf{y}^\mathsf{T}\mathbf{v}} \quad \text{and} \quad \overline{t}_j \leftarrow g^{s_j \mathbf{y}^\mathsf{T}\mathbf{v}} h^{\mathbf{y}^\mathsf{T}\mathbf{e}_j} \quad \text{and} \quad \overline{u}_j \leftarrow \prod_{k \in [\ell']} (\overline{z}^2_{j,k})^{y_k}.$$

  4. For each $i \in [\rho]$, compute $r_i \leftarrow H\big(\prod_{j \in S_i} \overline{t}_j^{2\alpha_j}\big)$ and $\pi_i \leftarrow \{(j, \overline{t}_j, \overline{u}_j)\}_{j \in S_i}$.
  5. The challenger gives r to $\mathcal{A}$ and receives a set $I \subseteq [\rho]$.
  6. The challenger replies with $\sigma = (\mathsf{s}, \overline{c})$ and the set $\{\pi_i\}_{i \in I}$.

- Hyb_2: Same as Hyb_1, except when responding to challenge queries, the challenger samples $\mathbf{y} \xleftarrow{R} \mathbb{Z}^{\ell'}_{Np'q'}$ instead of $\mathbf{y} \xleftarrow{R} \mathbb{Z}^{\ell'}_{\lfloor N^2/4 \rfloor}$.

- Hyb_3: Same as Hyb_2, except when responding to the challenge queries, the challenger first samples $\mathbf{y}' \xleftarrow{R} \mathbb{Z}^{\ell'}_{p'q'}$ and $\hat{\mathbf{y}}' \xleftarrow{R} \mathbb{Z}^{\ell'}_N$. It defines $\mathbf{y} \in \mathbb{Z}^{\ell'}_{Np'q'}$ to be the vector where $\mathbf{y} = \mathbf{y}' \pmod{p'q'}$ and $\mathbf{y} = \hat{\mathbf{y}}' \pmod{N}$.

- Hyb_4: Same as Hyb_3, except when responding to challenge queries, the challenger samples s and $\mathbf{y}'$ as in Hyb_3. Next, for each $i \in [\rho]$, it samples $\hat{\omega}_i \xleftarrow{R} \mathbb{Z}_N$, and for $j \in S_i$, it samples $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_N$ subject to the constraint that $\sum_{j \in S_i} \alpha_j \hat{y}'_j = \hat{\omega}_i$. For all of the remaining indices $j \in [\ell'] \setminus \bigcup_{i \in I} S_i$, sample $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_N$.

- Hyb_5: Same as Hyb_4, except when responding to the challenge queries, the challenger first samples $r \xleftarrow{R} \{0,1\}^\rho$. Then, after sampling s and $\mathbf{y}'$, it computes for each $i \in [\rho]$, $\omega_i \leftarrow \sum_{j \in S_i} \alpha_j s_j \pmod{p'q'}$. Then, it samples $\hat{\omega}_i \xleftarrow{R} \mathbb{Z}_N$ conditioned on $r_i = H\big((g^{\omega_i (\mathbf{y}')^\mathsf{T} \mathbf{v}} h^{\hat{\omega}_i})^2\big)$. It uses the same rejection sampling procedure as in $\mathcal{S}_2$ to sample $\hat{\omega}_i$.

- Hyb_6: Same as Hyb_5, except when responding to the challenge queries, the challenger samples $\hat{\mathbf{y}}' \in \mathbb{Z}_N^{\ell'}$ *after* it receives the challenge set. On each query, after the challenger receives the set $I \subseteq [\rho]$, for each $i \in I$, it samples $\hat{\mathbf{y}}'$ as follows:

  - For each $i \in I$, let $\omega_i \leftarrow \sum_{j \in S_i} \alpha_j s_j \pmod{p'q'}$. Then, sample $\hat{\omega}_i \xleftarrow{R} \mathbb{Z}_N$ conditioned on $r_i = H\big((g^{\omega_i (\mathbf{y}')^\mathsf{T} \mathbf{v}} h^{\hat{\omega}_i})^2\big)$ using the same rejection sampling procedure as in Hyb_5. After sampling $\hat{\omega}_i$, sample $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_N$ for each $j \in S_i$ subject to the constraint that $\sum_{j \in S_i} \alpha_j \hat{y}'_j = \hat{\omega}_i$.
  - For all of the remaining indices $j \in [\ell'] \setminus \bigcup_{i \in I} S_i$, sample $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_N$.

  The remaining components are constructed as in Hyb_5. This is exactly the distribution in $\mathsf{ExptHide}[\mathcal{A}, \mathcal{S}, 1]$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output of $\mathsf{Hyb}_i(\mathcal{A})$ with adversary $\mathcal{A}$. In the following, we show that the output distribution on each pair of adjacent experiments is statistically indistinguishable (or identically distributed).

Lemma E.22. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \equiv \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. Since $\mathcal{S}_1(1^\lambda, 1^\rho)$ samples $\widetilde{\mathsf{crs}}$ using the same procedure as $\mathsf{Setup}(1^\lambda, 1^\rho)$, the output distributions of Hyb_0 and Hyb_1 are identical. □

Lemma E.23. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. Follows by the same argument as the proof of Lemma E.12. □

Lemma E.24. *For all adversaries $\mathcal{A}$, $\mathsf{Hyb}_2(\mathcal{A}) \equiv \mathsf{Hyb}_3(\mathcal{A})$.*

Proof. Follows by the Chinese remainder theorem (as in the proof of Lemma E.13). □

Lemma E.25. *If G is a secure PRG, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_3(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_4(\mathcal{A})$.*

Proof. In Hyb_3, the challenger samples $\hat{\mathbf{y}}' \xleftarrow{R} \mathbb{Z}_N^{\ell'}$ while in Hyb_4, the challenger samples $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_N$ subject to the constraint that $\sum_{j \in S_i} \alpha_j \hat{y}'_j = \hat{\omega}_i$ where $\hat{\omega}_i \xleftarrow{R} \mathbb{Z}_N$ for $j \in S_i$ and $i \in [\rho]$. For the indices $j \in [\ell'] \setminus \bigcup_{i \in I} S_i$, $\hat{y}'_j \xleftarrow{R} \mathbb{Z}_N$. These distributions are identical as long as there exists some $j \in S_i$ where $\alpha_j \in \mathbb{Z}_N^*$ for each $i \in [\rho]$. Since G is a secure PRG, this holds with overwhelming probability (by a similar argument as in the proof of Claim C.11). □

Lemma E.26. *If $\mathcal{H}$ is a universal hash, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_4(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_5(\mathcal{A})$.*

Proof. In Hyb_4, except the challenger samples $\hat{\mathbf{y}} \in \mathbb{Z}_N^{\ell'}$ first in Hyb_4 while it samples $r \in \{0,1\}^\rho$ first in Hyb_5. We show that these two distributions are statistically indistinguishable. In Hyb_4, each challenge bit r_i satisfies

$$r_i = H\left(\prod_{j \in S_i} \overline{t}_j^{2\alpha_j}\right) = H\left(\prod_{j \in S_i} (g^{s_j \mathbf{y}^\mathsf{T} \mathbf{v}} h^{\mathbf{y}^\mathsf{T} \mathbf{e}_j})^{2\alpha_j}\right) = H((g^{\omega_i (\mathbf{y}')^\mathsf{T} \mathbf{v}} h^{\hat{\omega}_i})^2), \tag{E.3}$$

where $\omega_i = \sum_{j \in S_i} \alpha_j s_j \pmod{p'q'}$ and $\omega_i' = \sum_{j \in S_i} \alpha_j \hat{y}_j' \pmod{N}$, since g generates a group of order $p'q'$ and h generates a group of order N. Now, in Hyb_4, each $\hat{\omega}_i$ is uniform over $\mathbb{Z}_N$, and r_i is derived from Eq. (E.3), while in Hyb_5, $r_i \xleftarrow{\mathrm{R}} \{0,1\}$, and $\hat{\omega}_i$ is sampled uniformly from $\mathbb{Z}_N$ subject to Eq. (E.3). Using an analogous argument to Lemma E.14, these two distributions are statistically indistinguishable. □

Lemma E.27. *If G is a secure PRG, then for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_5(\mathcal{A}) \overset{s}{\approx} \mathsf{Hyb}_6(\mathcal{A})$.*

Proof. The challenger samples $\widetilde{\mathsf{crs}}$ identically in the two experiments, so N, g, h, $\mathbf{v}$, $s_1, \ldots, s_{\ell'}$ are identically distributed in the two experiments. so it suffices to consider its responses to the challenge queries. In both experiments, on each query, the challenger starts by sending the adversary a random string $r \xleftarrow{\mathrm{R}} \{0,1\}^\rho$, and the adversary replies with a set $I \subseteq [\rho]$. The challenger then replies with a commitment $\sigma = (\mathsf{s}, \overline{c})$ and a set of proofs $\{\pi_i\}_{i \in I}$, where $\pi_i = \{(j, \overline{t}_j, \overline{u}_j)\}_{j \in S_i}$. We show that the challenger's responses are statistically indistinguishable in the two experiments. To this end, we define the following variables:

- For $i \in [\rho]$, let $\hat{\mathbf{y}}^{(i)} \in \mathbb{Z}_N^\ell$ be the vector where $\hat{y}_j^{(i)} = \hat{y}'_{j+\ell\cdot(i-1)} \in \mathbb{Z}_N$. In other words, $(\hat{\mathbf{y}}')^\mathsf{T} = [\ (\hat{\mathbf{y}}^{(1)})^\mathsf{T} \mid \cdots \mid (\hat{\mathbf{y}}^{(\rho)})^\mathsf{T}\ ] \in \mathbb{Z}_N^{\ell'}$.
- For $i \in [\rho]$, define $\boldsymbol{\alpha}^{(i)} = (\alpha_1^{(i)}, \ldots, \alpha_\ell^{(i)})^\top \in \mathbb{Z}_N^\ell$ as follows:

$$\alpha_j^{(i)} = \begin{cases} \alpha_{j+\ell\cdot(i-1)} & \text{if } j \in \hat{S}_i \\ 0 & \text{otherwise.} \end{cases}$$

We now consider the different components in the two experiments:

- The string r is independently and uniformly sampled from $\{0,1\}^\rho$ in both experiments.
- The seed s is independently and uniformly sampled from $\{0,1\}^\kappa$ in both experiments, so the seed s, the sets $S_1, \ldots, S_\rho$, and the vector $\boldsymbol{\alpha}$ are identically distributed as well.
- The vector $\mathbf{y}'$ is sampled uniformly from $\mathbb{Z}_{p'q'}^{\ell'}$ in both experiments. Since g generates a group of order $p'q'$, the commitment $\overline{c} = g^{\mathbf{y}^\mathsf{T}\mathbf{v}} = g^{(\mathbf{y}')^\mathsf{T}\mathbf{v}}$ is identically distributed in Hyb_5 and Hyb_6.
- For all $i \in I$, both experiments set $\omega_i \leftarrow \sum_{j \in S_i} \alpha_j s_j \pmod{p'q'}$ and sample $\hat{\omega}_i \xleftarrow{\mathrm{R}} \mathbb{Z}_N$ subject to $r_i = H\big((g^{\omega_i (\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{\omega}_i})^2\big)$. This means that the vectors $\hat{\mathbf{y}}^{(i)}$ for $i \in I$ are identically distributed in the two experiments. Then, for $j \in S_i$, we have that $\overline{t}_j = g^{s_i (\mathbf{y}')^\mathsf{T}\mathbf{v}} h^{\hat{y}'_j}$, so the components $\overline{t}_j$ for $j \in S_i$ and $i \in I$ are identically distributed.

It suffices to show that the remaining components $\{\overline{u}_j\}_{j\in S_i}$ for $i \in I$ are drawn from statistically indistinguishable distributions (from the view of the adversary). First, let

$$J = \{j \in S_i \text{ for some } i \in I \mid j \in [\ell']\}$$

be the set of indices that appear in some set S_i for $i \in I$. Since $|S_i| = \lambda$ and the S_i's are pairwise disjoint, $|J| = \lambda\,|I| \leq \lambda\rho = \ell/3$.

It suffices to consider the case where $\overline{\mathbf{z}}_j \in \mathbb{Z}^*_{N^2}$ for all $j \in [\ell']$. Otherwise, both experiments output $\perp$. This means $\overline{z}^2_{j,k} \in \mathbb{NR}_{2N} \times \mathbb{H}$, and correspondingly, we can write $\overline{z}^2_{j,k} = g^{z_{j,k}} h^{\hat{z}_{j,k}}$ for some $z_{j,k} \in \mathbb{Z}_{p'q'}$ and $\hat{z}_{j,k} \in \mathbb{Z}_N$. Now, for all $j \in S_i$ and $i \in I$, the challenger (in both experiments) computes $\overline{u}_j$ as

$$\overline{u}_j = \prod_{k\in[\ell']} (\overline{z}^2_{j,k})^{y_k} = \prod_{j\in[\ell']} g^{z_{j,k}y_k} h^{\hat{z}_{j,k}y_k} = g^{(\mathbf{y}')^\mathsf{T}\mathbf{z}_j} h^{(\hat{\mathbf{y}}')^\mathsf{T}\hat{\mathbf{z}}_j}.$$

Since $\mathbf{y}'$ is identically distributed in Hyb_5 and Hyb_6, it suffices to consider the distribution of $\overline{u}_j$ in the $\mathbb{H}$ subgroup, or equivalently, the distribution of $(\hat{\mathbf{y}}')^\mathsf{T}\hat{\mathbf{z}}_j$ over $\mathbb{Z}_N$. Let $\hat{\mathbf{Z}} \in \mathbb{Z}_N^{\ell'\times\ell'}$ be the matrix whose columns are $\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_{\ell'}$. Let $\hat{\mathbf{Z}}' \in \mathbb{Z}_N^{\ell'\times\lambda|I|}$ be the submatrix of $\hat{\mathbf{Z}}$ formed by taking only the columns in $\hat{\mathbf{Z}}$ indexed by the set J. In particular, the values of $h^{(\hat{\mathbf{y}}')^\mathsf{T}\hat{\mathbf{Z}}'} \in \mathbb{H}^{\lambda|I|}$ precisely coincide with the components in the $\mathbb{H}$-subgroup of $\overline{u}_j$ for $j \in S_i$ and $i \in I$. Thus, it suffices to show that for these indices $j \in S_i$ and $i \in I$, the distributions of $(\hat{\mathbf{y}}')^\mathsf{T}\hat{\mathbf{Z}}' \in \mathbb{Z}_N^{\lambda|I|}$ are statistically indistinguishable in the two experiments. First, for $i \in [\rho]$, let $\hat{\mathbf{Z}}^{(i)} \in \mathbb{Z}_N^{\ell\times\lambda|I|}$ be matrices such that

$$\hat{\mathbf{Z}}' = \begin{bmatrix} \hat{\mathbf{Z}}^{(1)} \\ \vdots \\ \hat{\mathbf{Z}}^{(\rho)} \end{bmatrix},$$

This means that

$$(\hat{\mathbf{y}}')^\mathsf{T}\hat{\mathbf{Z}}' = \sum_{i\in[\rho]} \left((\hat{\mathbf{y}}^{(i)})^\mathsf{T}\hat{\mathbf{Z}}^{(i)}\right) \in \mathbb{Z}_N^{\lambda|I|}. \tag{E.4}$$

By definition, $\hat{\mathbf{Z}}$ is a *fixed* matrix (determined by the CRS and the adversary's public key) and independent of $\hat{\mathbf{y}}'$. Since each $\hat{\mathbf{y}}^{(i)}$ is sampled independently, we can consider each term individually in this summation:

- If $i \in I$, then as argued above, $\hat{\mathbf{y}}^{(i)}$ is identically distributed in Hyb_5 and Hyb_6, and correspondingly, so is the product $(\hat{\mathbf{y}}^{(i)})^\mathsf{T}\hat{\mathbf{Z}}^{(i)}$.
- If $i \notin I$, then the $\hat{\mathbf{y}}^{(i)}$ in the two experiments are sampled from distinct distributions. In Hyb_5, $\hat{\mathbf{y}}^{(i)}$ is uniform over $\mathbb{Z}_N^\ell$ subject to $\hat{\omega}_i = \sum_{j\in[\ell]} \alpha_j^{(i)} \hat{y}_j^{(i)}$, while in Hyb_6, $\hat{\mathbf{y}}^{(i)}$ is uniform over $\mathbb{Z}_N^\ell$. By the Chinese remainder theorem, we can sample (and analyze) the $\mathbb{Z}_p$ and $\mathbb{Z}_q$ components of $\hat{\mathbf{y}}^{(i)}$ independently.

  Define $\hat{\mathbf{Z}}_p^{(i)}$, $\hat{\mathbf{Z}}_q^{(i)}$, $\boldsymbol{\alpha}_p^{(i)}$, and $\boldsymbol{\alpha}_q^{(i)}$ as follows:

$$\begin{aligned} \hat{\mathbf{Z}}_p^{(i)} &= \hat{\mathbf{Z}}^{(i)} \pmod p \qquad & \boldsymbol{\alpha}_p^{(i)} &= \boldsymbol{\alpha}^{(i)} \pmod p \\ \hat{\mathbf{Z}}_q^{(i)} &= \hat{\mathbf{Z}}^{(i)} \pmod q \qquad & \boldsymbol{\alpha}_q^{(i)} &= \boldsymbol{\alpha}^{(i)} \pmod q. \end{aligned}$$

By Claim C.11 (generalized to $\mathbb{Z}_N$ via the Chinese remainder theorem), we have that with overwhelming probability, $\boldsymbol{\alpha}_p^{(i)} \notin \mathrm{span}(\hat{\mathbf{Z}}_p^{(i)})$ and $\boldsymbol{\alpha}_q^{(i)} \notin \mathrm{span}(\hat{\mathbf{Z}}_q^{(i)})$. By Claim C.12, this means that the distribution of $(\hat{\mathbf{y}}^{(i)})^\mathsf{T}\hat{\mathbf{Z}}_p^{(i)} \pmod p$ in Hyb_5 (where $\hat{\mathbf{y}}^{(i)} \pmod p$ is uniform subject to a linear constraint $\boldsymbol{\alpha}_p^{(i)} \notin \mathrm{span}(\hat{\mathbf{Z}}_p^{(i)})$) is statistically indistinguishable from its distribution in Hyb_6 (where $\hat{\mathbf{y}}^{(i)} \pmod p$ is uniform). By the same argument, the distributions of $(\hat{\mathbf{y}}^{(i)})^\mathsf{T}\hat{\mathbf{Z}}_q^{(i)} \pmod q$ in Hyb_5 and Hyb_6 are also statistically indistinguishable. By the Chinese remainder theorem and a union bound, this means that the product $(\hat{\mathbf{y}}^{(i)})^\mathsf{T}\hat{\mathbf{Z}}^{(i)} \in \mathbb{Z}_N^{\lambda|I|}$ is statistically indistinguishable in the two experiments.

Since every term in Eq. (E.4) is either statistically indistinguishable or identically distributed in the two experiments, we conclude that $\hat{\mathbf{Z}}'\hat{\mathbf{y}}'$ is also statistically indistinguishable in the two experiments. Correspondingly, this means that the components $\overline{u}_j$ for $j \in S_i$ and $i \in I$ are also statistically indistinguishable in the two experiments. □

Since each consecutive pair of hybrid experiments is statistically indistinguishable (or identically distributed), the theorem follows. □